Lecture Notes in Artificial Intelligence 5378

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

FoLLI Publications on Logic, Language and Information

R. Ramanujam   Sundar Sarukkai (Eds.)

Logic and Its Applications

Third Indian Conference, ICLA 2009
Chennai, India, January 7-11, 2009
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

R. Ramanujam
Institute of Mathematical Sciences
CIT Campus, Chennai 600113, India
E-mail: jam@imsc.res.in

Sundar Sarukkai
National Institute of Advanced Studies
Indian Institute of Science Campus, Bangalore 560012, India
E-mail: sarukkai@nias.iisc.erenet.in

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2.3, I.2, F.1-4, D.2.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN	0302-9743
ISBN-10	3-540-92700-X Springer Berlin Heidelberg New York
ISBN-13	978-3-540-92700-6 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12591718 06/3180 5 4 3 2 1 0

Preface

This volume contains the papers presented at ICLA 2009: Third Indian Conference on Logic and Its Applications (ICLA) held at the Institute of Mathematical Sciences, Chennai, Januay 7–11, 2009.

The ICLA series aims to bring together researchers from a wide variety of fields that formal logic plays a significant role in, along with mathematicians, philosophers, computer scientists and logicians studying foundations of formal logic in itself. A special feature of this conference are studies in systems of logic in the Indian tradition, and historical research on logic. The biennial conference is organized by the Association for Logic in India.

The papers in the volume span a wide range of themes. We have contributions to algebraic logic and set theory, combinatorics and philosophical logic. Modal logics, with applications in computer science and game theory, are discussed. Not only do we have papers discussing connections between ancient logical systems with modern ones, but also those offering computational tools for experimenting with such systems. It is hoped that ICLA will act as a platform for such dialogues arising from many disciplines, using formal logic as its common language.

Like the previous conferences (IIT-Bombay; January 2005 and 2007) and (Jadavpur University, Kolkata; January 2007), the third conference also manifested this confluence of several disciplines. As in the previous years, we were fortunate to have a number of highly eminent researchers giving plenary talks. It gives us great pleasure to thank Johan van Benthem, Rajeev Goré, Joel Hamkins, Johann Makowsky, Rohit Parikh, Esko Turunen and Moshe Vardi for agreeing to give invited talks and for contributing to this volume.

The Programme Committee, with help from many external reviewers, put in a great deal of hard work to select papers from the submissions. We express our gratitude to all members for doing an excellent job and thank all the reviewers for their invaluable help.

ICLA 2009 included two pre-conference workshops: one on Algebraic Logic coordinated by Mohua Banerjee (IIT Kanpur) and Mai Gehrke (Radboud Universiteit, Nijmegen), and another on Logics for Social Interaction coordinated by Sujata Ghosh (ISI Kolkata), Eric Pacuit (Stanford University) and R. Ramanujam (IMSc Chennai). We thank the organizers as well as the speakers in the workshops for contributing so significantly to the programme.

The conference was held at the Institute of Mathematical Sciences (IMSc), Chennai. We thank IMSc and the Organizing Committee for taking on the responsibility. Special thanks are due to Sunil Simon (IMSc) for help in preparation of this volume. The Easychair system needs special mention, for its tremendous versatility.

We also thank the Editorial Board of the FoLLI series and Springer for publishing this volume.

October 2008

R. Ramanujam
Sundar Sarukkai

Organization

Institute of Mathematical Sciences, Chennai

Programme Chairs

R. Ramanujam
Sundar Sarukkai

Programme Committee

Natasha Alechina
Mohua Banerjee
Agata Ciabattoni
Jonardon Ganeri
Mai Gehrke
Kevin Kelly
Kamal Lodaya
Benedikt Loewe
Larry Moss
Ranjan Mukhopadhyay
Eric Pacuit
Arun Pujari
N. Raja
K. Ramasubramanian
Mark Siderits

Local Organization

Sujata Ghosh
S. Krishna
Sunil Simon
Arindama Singh
S.P. Suresh

Reviewers

Natasha Alechina
Horacio Arlo-Costa
Mohua Banerjee
Walter Carnielli
Mihir K. Chakraborty
Anna Chernilovskaya
Agata Ciabattoni
Pierre-Louis Curien

Anuj Dawar
Paul Dekker
Antonio Di Nola
Hans van Ditmarsch
Didier Dubois
Christian Fermueller
Jonardon Ganeri
Mai Gehrke
Sujata Ghosh
Valentin Goranko
Joanna Grygiel
Wiebe van der Hoek
Tomohiro Hoshi
Gerard Huet
Juhani Karhumaki
Kevin Kelly
Benny Kenkireth
Amba Kulkarni
Tiago de Lima
Kamal Lodaya
Iris Loeb
Benedikt Loewe
George Metcalfe
Pabitra Mitra
Thomas Möller
Luca Motto Ros
Larry Moss
Ranjan Mukhopadhyay
Hiroakira Ono
Martin Otto
Wojciech Plandowski
Lech Polkowski
Eric Pacuit
Arun Pujari
N. Raja
R. Ramanujam
K. Ramasubramanian
Jan-Willem Romeijn
Prahlad Sampath
Sundar Sarukkai
Mark Siderits
Carsten Sinz
S.P. Suresh
Kazushige Terui
Joerg Tuske
Pawel Urzyczyn
Gregory Wheeler
Eva Wilhelmus
Stefan Woltran
Anna Zamansky

Table of Contents

Decisions, Actions, and Games: A Logical Perspective

Johan van Benthem

Amsterdam & Stanford
http://staff.science.uva.nl/∼johan

1 Introduction: Logic and Games

Over the past decades, logicians interested in rational agency and intelligent interaction studied major components of these phenomena, such as knowledge, belief, and preference. In recent years, standard 'static' logics describing information states of agents have been generalized to *dynamic* logics describing actions and events that produce information, revise beliefs, or change preferences, as explicit parts of the logical system. [22], [1], [12] are up-to-date accounts of this dynamic trend (the present paper follows Chapter 9 of the latter book). But in reality, concrete rational agency contains all these dynamic processes entangled. A concrete setting for this entanglement are *games* – and this paper is a survey of their interfaces with logic, both static and dynamic. Games are intriguing also since their analysis brings together two major streams, or tribal communities: 'hard' mathematical logics of computation, and 'soft' philosophical logics of propositional attitudes. Of course, this hard/soft distinction is spurious, and there is no natural border line between the two sources: it is their congenial mixture that makes current theories of agency so lively.

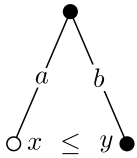
We will discuss both *statics*, viewing games as fixed structures representing all possible runs of some process, and the *dynamics* that arises when we make things happen on such a 'stage'. We start with a few examples showing what we are interested in. Then we move to a series of standard logics describing static game structure, from moves to preferences and epistemic uncertainty. Next, we introduce dynamic logics, and see what they add in scenarios with information update and belief revision where given games can change as new information arrives. This paper is meant to make a connection. It is not a full treatment of logical perspectives on games, for which we refer to [13].

2 Decisions, Practical Reasoning, and 'Solving' Games

Action and Preference. Even the simplest scenarios of practical reasoning about agents involve a number of notions at the same time:

Example 1 (One single decision). An agent has two alternative courses of action, but prefers one outcome to the other:

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 1–22, 2009.



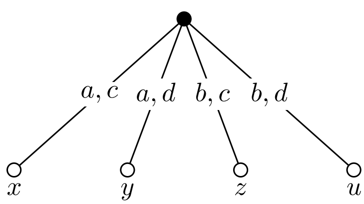
A proto-typical form of reasoning here would be the ‘Practical Syllogism’:

1. the agent can do both a and b,
2. the agent prefers the result of a over the result of b, and therefore,
3. the agent will do (or maybe: should do?) b.

This predictive inference, or maybe requirement, is in fact the basic notion of *rationality* for agents throughout a vast literature in philosophy, economics, and many other fields. It can be used to predict behaviour beforehand, or rationalize observed behaviour afterwards.

Adding Beliefs. In decision scenarios, *preference* crucially occurs intertwined with *action*, and a reasonable way of taking the conclusion is, not as knowledge ruling out courses of action, but as supporting a *belief* that the agent will take action b: the latter event is now more plausible than the world where she takes action a. Thus, modeling even very simple decision scenarios involves logics of different kinds. Beliefs come in even more strongly when one models uncertainty about possible states of nature, and one is told to choose the action with the highest *expected value*, a probabilistically weighted sum of utility values for the various outcomes. The probability distribution over states of nature represents *beliefs* we have about the world, or the behaviour of an opponent. Here is a yet simpler scenario:

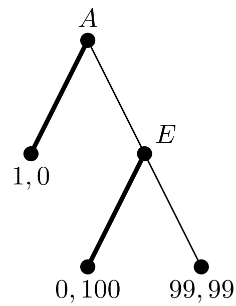
Example 2 (Deciding with an external influence). Nature has two moves c, d, and the agent must now consider combined moves:



Now, the agent might already have good reasons to think that Nature’s move c is more plausible than move d. This turns the outcomes into a ‘epistemic-doxastic model’ [7]: the epistemic range has 4 worlds, but the most plausible ones are just: x, z, while an agent’s preference might now just refer to the latter area.

Multi-agent Decision: ‘Solving’ Games by Backward Induction. In a multi-agent setting, behaviour is locked in place by *mutual expectations*. This requires an interactive decision dynamics, and standard game solution procedures like *Backward Induction* do exactly that:

Example 3 (Reasoning about interaction). In the following game tree, players' preferences are encoded in the utility values, as pairs '(value of A, value for E)'. Backward Induction tells player E to turn left when she can, just like in our single decision case, which gives A the belief that this would happen, and so, based on this belief about his counter-player, A should turn left at the start:



Why should players act this way? The reasoning is again a mixture of all notions so far. A turns left since she believes that E will turn left, and then her preference is for grabbing the value 1. Thus, practical reasoning intertwines action, preference, and belief.

Here is the rule which drives all this, at least when preferences are encoded numerically:

Definition 1 (Backward Induction algorithm). *Starting from the leaves, one assigns values for each player to each node, using the rule:*

> *Suppose E is to move at a node, and all values for daughters are known. The E-value is the maximum of all the E-values on the daughters, the A-value is the minimum of the A-values at all E-best daughters. The dual calculation for A's turns is completely analogous.*

This rule is so obvious that it never raises objections when taught, and it is easy to apply, telling us what players' best course of action would be [27]. And yet, it is packed with various assumptions. We will perform a 'logical deconstruction' of the underlying reasoning later on, but for now, just note the following features:

1. the rule assumes that the situation is viewed in the same way by both players: since the calculations are totally similar,
2. the rule assumes worst-case behaviour on the part of one's opponents, since we take a minimum of values in case it is not our turn,
3. the rule changes its interpretation of the values: at leaves they encode plain utilities, while higher up in the game tree, they represent expected utilities.

Thus, despite its numerical trappings, Backward Induction is an inductive mechanism for generating a *plausibility order* among histories, and hence, it relates all notions that we are interested in. There has been a lot of work on

'justifying' this solution method. Personally, I am not committed to this particular style of solving games, but understanding what Backward Induction does is a logically rich subject, which can be pursued in many ways.

But for now, we step back, and look at what 'logic of games' would involve *ab initio*, even without considering any preferences at all. So, let us first consider pure action structure, because even that has a good deal of logic to it, which can be brought out as such. We will add further preferential and epistemic structure toward more realistic games in due course.

3 Games and Process Equivalence

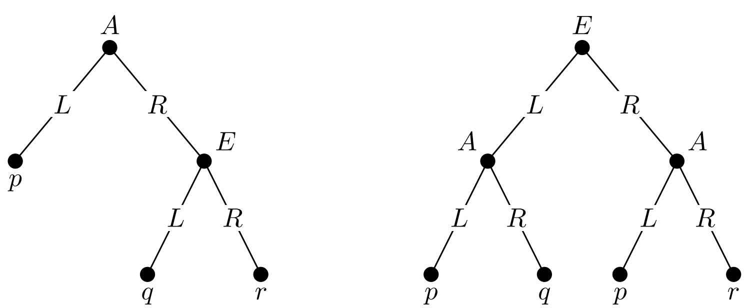
One can view extensive games as multi-agent processes that can be studied just like any process in logic and computer science, given the right logical language. Technically, such structures are models for a poly-modal logic in the following straightforward sense:

Definition 2 (Extensive games). *An extensive game form is defined to be a tree* $M = (\boldsymbol{NODES}, \boldsymbol{MOVES}, \boldsymbol{turn}, \boldsymbol{end}, \boldsymbol{V})$ *which is a modal model with binary transition relations taken from the set* $\boldsymbol{MOVES}$ *pointing from parent to daughter nodes. Also, intermediate nodes have unary proposition letters* $\boldsymbol{turn}_i$ *indicating the unique player whose turn it is, while* **end** *marks end nodes without further moves. The valuation* $\boldsymbol{V}$ *for proposition letters may also interpret other relevant predicates at nodes, such as utility values for players or more external properties of game states.*

But do we really just want to jump on board of this analogy, comfortable as it is to a modal logician? Consider the following fundamental issue of invariance in process theories. At which level do we want to operate in the logical study of games, or in Clintonesque terms:

When are two games are the same?

Example 4 (The same game, or not?). As a simple example that is easy to remember, consider the following two games:



Are these the same? As with general processes in computer science, the answer crucially depends on our level of interest in the details of what is going on:

1. If we focus on turns and moves, then the two games are not equivalent.
   For they differ in 'protocol' (who gets to play first) and in choice structure. For instance, the first game, but not the second has a stage where it is up to E to determine whether the outcome is q or r.
   This is indeed a natural level for looking at game, involving local actions and choices, as encoded in modal bisimulations – and the appropriate language will be a standard modal one. But one might also want to call these games equivalent in another sense: looking at achievable outcomes only, and players powers for controlling these:

2. If we focus on outcome powers only, then the two games are equivalent.
   The reason is that, regardless of protocol and local choices, players can force the same sets of eventual outcomes across these games, using strategies that are available to them:
   A can force the outcome to fall in the sets $\{p\}, \{q, r\}$,
   E can force the outcome to fall in the sets $\{p, q\}, \{p, r\}$.
   In the left-hand tree, A has 2 strategies, and so does E, yielding the listed sets. In the right-hand tree, E has 2 strategies, while A has 4: *LL, LR, RL* and *RR*. Of these, *LL* yields the outcome set $\{p\}$, and *RR* yields $\{q, r\}$. But *LR, RL* guarantee only supersets $\{p, r\}, \{q, p\}$ of $\{p\}$: i.e., weaker powers. Thus the same 'control' results in both games.

We will continue on extensive games, but the coarser power level is natural, too. It is like 'strategic forms' in game theory, and it fits well with 'logic games' [8]:

Remark 1 (Game equivalence as logical equivalence). In an obvious sense, the two games in the preceding example represent the two sides of the following valid logical law

$$p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r) \qquad \textit{Distribution}$$

Just read conjunction and disjunction as choices for different players. In a global input-output view on games, *Distribution* switches scheduling order without affecting players' powers.

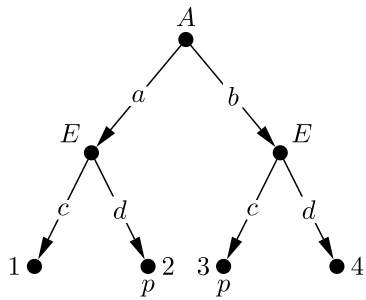
4 Basic Modal Action Logic of Extensive Games

Basic Modal Logic. On extensive game trees, a standard modal language works as follows:

Definition 3 (Modal game language and semantics). *Modal formulas are interpreted at nodes s in game trees M. Labeled modalities $\langle a\rangle\varphi$ express that some move a is available leading to a next node in the game tree satisfying φ. Proposition letters true at nodes may include special-purpose constants for typical game structure, such as markings for turns and end-points, but also arbitrary local properties.*

In particular, modal operator combinations now describe potential interaction:

Example 5 (Modal operators and strategic powers). Consider a simple 2-step game like the following, between two players A, E:



Player E clearly has a strategy making sure that a state is reached where p holds. And this feature of the game is directly expressed by the modal formula $[a]\langle d\rangle p \wedge [b]\langle c\rangle p$.

Letting *move* be the union of all moves available to players, a modal operator combination $[move_A]\langle move_E\rangle\varphi$ says that, at the current node, player E has a strategy for responding to A's initial move which ensures that the property expressed by φ results after two steps.[1]

Excluded Middle and Determinacy. Extending this observation to extensive games up to some finite depth k, and using alternations $\Box\Diamond\Box\Diamond\cdots$ of modal operators up to length k, we can express the existence of winning strategies in fixed finite games. Indeed, given this connection, with finite depth, standard logical laws have immediate game-theoretic import. In particular, consider the valid *law of excluded middle* in the following modal form

$$\Box\Diamond\Box\Diamond\cdots\varphi \vee \neg\Box\Diamond\Box\Diamond\cdots\varphi$$

or after some logical equivalences, pushing the negation inside:

$$\Box\Diamond\Box\Diamond\cdots\varphi \vee \Diamond\Box\Diamond\Box\cdots\neg\varphi,$$

where the dots indicate the depth of the tree. Here is its game-theoretic import:

Fact 4. *Modal excluded middle expresses the determinacy of finite games.*

Determinacy is the key property that *one of the two players has a winning strategy.* This need not be true in infinite games (players cannot both have one, but maybe neither has).

[1] One can also express existence of ‘winning strategies’, ‘losing strategies’, and so on.

Zermelo's Theorem. This brings us to perhaps the oldest game-theoretic result proved in mathematics, even predating Backward Induction, proved by Ernst Zermelo in 1913:

Theorem 5. *Every finite zero-sum 2-player game is determined.*

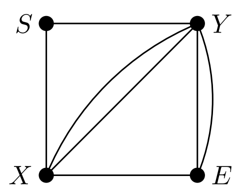
Proof. Here is a simple algorithm determining the player having the winning strategy at any given node of a game tree of this finite sort. It works bottom-up through the game tree. First, colour those end nodes *black* that are wins for player A, and colour the other end nodes *white*, being the wins for E. Then extend this colouring stepwise as follows:

> If all children of node s have been coloured already, do one of the following:
>
> 1. if player A is to move, and at least one child is black: colour s *black*; if all children are white, colour s *white*
> 2. if player E is to move, and at least one child is white: colour s *white*; if all children are black, colour s *black*

This procedure eventually colours all nodes black where player A has a winning strategy, making those where E can win white. The reason for its correctness is easy to see.

Zermelo's Theorem is widely applicable. Consider the following Teaching Game:

Example 6 (Teaching, the grim realities). A Student located at position S in the next diagram wants to reach the escape E below, while the Teacher wants to prevent him from getting there. Each line segment is a path that can be traveled. In each round of the game, the Teacher cuts one connection, anywhere, while the Student can, and must travel one link still open to him at his current position:



Education games like this arise on any graph with single or multiple lines.

We now have an explanation why Student or Teacher has a winning strategy: the game is two-player zero sum and of finite depth – though it need not have an effective solution. Zermelo's Theorem implies that in Chess, one player has a winning strategy, or the other a non-losing one, but a century later, we do not know which: the game tree is too large.

5 Fixed-Point Languages for Equilibrium Concepts

A good test for logics is their expressive power in representing proofs of significant results. Now our modal language cannot express the *generic character* of the Zermelo solution. Here is what the colouring algorithm really says. Starting from atomic predicates win_i at end nodes indicating which player has won, we inductively defined predicates WIN_i ('player i has a winning strategy at the current node') through the following recursion:

$$WIN_i \leftrightarrow (end \wedge win_i) \vee (turn_i \wedge \langle E \rangle\, WIN_i) \vee (turn_j \wedge [A]\, WIN_i)$$

Here E is the union of all available moves for player i, and A that of all moves for the counter-player j. This schema is an inductive definition for the predicate WIN_i, which we can also write as a *smallest fixed-point* expression in an extended modal language:

Fact 6. *The Zermelo solution is definable as follows in the modal μ-calculus:*

$$WIN_i = \mu p[(end \wedge win_i) \vee (turn_i \wedge \langle E \rangle p) \vee (turn_j \wedge [A]p)]^2$$

Here the formula on the right-hand side belongs to the *modal μ-calculus*, an extension of the basic modal language with operators for *smallest* (and *greatest*) fixed-points defining inductive notions. This system was originally invented to increase the power of modal logic as a process theory. We refer to the literature for details, cf. [20]. Fixed-points fit well with strategic equilibria, and the μ-calculus has further uses in games.

Definition 7 (Forcing modalities). Forcing modalities *are interpreted as follows in extensive game models as defined earlier: $M, s \models \{i\}\varphi$ iff player i has a strategy for the sub-game starting at s which guarantees that only nodes will be visited where φ holds, whatever the other player does.*

Forcing talk is widespread in games, and it is an obvious target for logical formalization:[3]

Fact 8. *The modal μ-calculus can define forcing modalities.*

Proof. The formula $\{i\}\varphi = \mu p[(\varphi \wedge end) \vee (turn_i \wedge \langle E \rangle p) \vee (turn_j \wedge [A]p))]$ defines the existence of a strategy for i ensuring that proposition φ holds, whatever the other plays.

[2] Note that the defining schema only has *syntactically positive occurrences* of the predicate p.

[3] Note that $\{i\}\varphi$ talks about intermediate nodes, not just the end nodes of a game. The existence of a winning strategy for player i can then be formulated by restricting to endpoints: $\{i\}(end \rightarrow win_i)$.

But many other notions are definable. For instance, the recursion

$$COOP\varphi \leftrightarrow \mu p[(end \wedge \varphi) \vee (turn_i \wedge \langle E \rangle p) \vee (turn_j \wedge \langle A \rangle p)]$$

defines the existence of a *cooperative outcome* φ, just by shifting modalities.[4]

Digression: From Smallest to Greatest Fixed-Points. The above modal fixed-point definitions reflect the equilibrium character of basic game-theoretic notions [27], reached through some process of iteration. In this general setting, which includes infinite games, we would switch from smallest to greatest fixed-points, as in the formula

$$\{i\}\varphi = \nu q[(\varphi \wedge (turn_i \wedge \langle move_i \rangle q) \vee (turn_j \wedge [move_j]q))].$$

This is also more in line with our intuitive view of strategies. The point is not that they are built up from below, but that they can be used as needed, and then remain at our service as pristine as ever the next time - the way we think of doctors. This is the modern perspective of *co-algebra* [29]. More generally, greatest fixed-points seem the best logical analogue to the standard equilibrium theorems from analysis that are used in game theory.

But Why Logic? This may be a good place to ask what is the point of logical definitions of game-theoretic notions? I feel that logic has the same virtues for games as elsewhere. Formalization of a practice reveals what makes its key notions tick, and we also get a feel for new notions, as the logical language has myriads of possible definitions. Also, the theory of expressive power, completeness, and complexity of our logics can be used for model checking, proof search, and other activities not normally found in game theory.

But there is also another link. Basic notions of logic *themselves* have a game character, such as argumentation, model checking, or model comparison. Thus, logic does not just *describe games*, it also *embodies games*. Pursuing the interface in this dual manner, the true grip of the logic and games connection becomes clear: cf. [13].

6 Dynamic Logics of Strategies

Strategies, rather than single moves, are protagonists in games, Moving them in focus requires an extension of modal logic to *propositional dynamic logic (PDL)* which describes structure and effects of imperative programs with operations of (a) sequential composition ;, (b) guarded choice $IF \cdots THEN \cdots ELSE \cdots$, and (c) guarded iterations $WHILE \cdots DO \cdots$:

[4] This fixed point can still be defined in propositional dynamic logic, using the formula $\langle(((turn_i)?;E) \cup ((turn_j)?;A))^*\rangle(end \wedge \varphi)$, – but we will only use the latter system later in the game setting.

Definition 9 (Propositional dynamic logic). *The* language of PDL *defines formulas and programs in a mutual recursion, with formulas denoting sets of worlds ('local conditions' on 'states' of the process), while programs denote binary transition relations between worlds, recording pairs of input and output states for their successful terminating computations. Programs are created from*

> *atomic actions ('moves') $a, b, \dots$ and tests $?\phi$ for arbitrary formulas ϕ, using the three operations of* ; *(interpreted as sequential composition),* $\cup$ *(non-deterministic choice) and* * *(non-deterministic finite iteration).*

Formulas are as in our basic modal language, but with modalities $[\pi]\phi$ saying that ϕ is true after every successful execution of the program π starting at the current world.

The logic *PDL* is decidable, and it has a transparent complete set of axioms for validity. This formalism can say a lot more about our preceding games. For instance, the *move* relation in our discussion of our first extensive game was really a union of atomic transition relations, and the pattern that we discussed for the winning strategy was as follows:

$$[a \cup b]\langle c \cup d\rangle p.$$

Strategies as Transition Relations. Game-theoretic strategies are partial transition functions defined on players' turns, given via a bunch of conditional instructions of the form "if she plays this, then I play that." More generally, strategies may be viewed as binary transition relations, allowing for non-determinism, i.e., more than one 'best move', like *plans* that agents have in interactive settings. A plan can be useful, even when it merely constrains my future moves. Thus, on top of the 'hard-wired' moves in a game, we get defined relations for players' strategies, and these definitions can often be given explicitly in a *PDL*-format.

In particular, in finite games, we can define an explicit version of the earlier forcing modality, indicating the strategy involved – without recourse to the modal μ-calculus:

Fact 10. *For any game program expression σ, PDL can define an explicit forcing modality $\{\sigma, i\}\varphi$ stating that σ is a strategy for player i forcing the game, against any play of the others, to pass only through states satisfying φ.*

The precise definition is an easy exercise (cf. [5]). Also, given strategies for both players, we should get to a unique history of a game, and here is how:

Fact 11. *Outcomes of running joint strategies σ, τ can be defined in PDL.*

Proof. The formula $[((?turn_E; \sigma) \cup (?turn_A; \tau))^*](end \to p)$ does the job.[5]

Also 'locally', PDL can define specific strategies. Take any finite game M with strategy σ for player i. As a relation, σ is a finite set of ordered pairs (s, t). Thus,

[5] Dropping the antecedent '$end \to$' here will describe effects of strategies at intermediate nodes.

it can be defined by a program union, if we first define these ordered pairs. To do so, assume we have an 'expressive' model M, where states s are definable in our modal language by formulas def_s.[6] Then we define transitions (s,t) by formulas $def_s; a; def_t$, with a being the relevant move:

Fact 12. *In expressive finite extensive games, all strategies are PDL-definable.*

Dynamic logic can also define strategies running over only part of a game, and their *combination.* The following modal operator describes the effect of such a partial strategy σ for player E running until the first game states where it is no longer defined:

$$\{\sigma, E\}\varphi = [(?turn_E; \sigma) \cup (?turn_A; move_A)*]\varphi^7$$

7 Preference Logic and Defining Backward Induction

Real games go beyond game forms by adding preferences for players over outcome states, or numerical utilities beyond 'win' and 'lose'. In this area, defining the Backward Induction procedure for solving extensive games, rather than computing binary Zermelo winning positions, has become a benchmark for game logics – and many solutions exist:

Fact 13. *The Backward Induction path is definable in modal preference logic.*

Solutions have been published by many logicians and game-theorists in recent years, cf. [21,25]. We do not state an explicit *PDL*-style solution here, but we give one version involving a *modal preference language* with this operator:

$\langle pref_i \rangle \varphi$: player i prefers some node where φ holds to the current one.

The following result from [18] defines the backward induction path as a unique relation σ: not by means of any specific modal formula in game models M, but rather via the following frame correspondence on finite structures:

Fact 14. *The BI strategy is definable as the unique relation σ satisfying the following axiom for all propositions P – viewed as sets of nodes –, for all players i:*

$$(turn_i \wedge \langle \sigma^* \rangle (end \wedge P)) \rightarrow [move_i]\langle \sigma^* \rangle (end \wedge \langle pref_i \rangle P).$$

Proof. The axiom expresses a form of rationality: at the current node, no alternative move for a player guarantees outcomes that are all strictly better than those ensuing from playing the current backward induction move. The proof is by induction on the game tree.

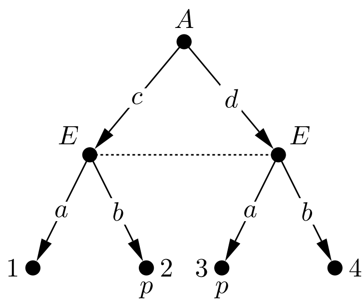
[6] This expressive power can be achieved: e.g., using temporal *past modalities* involving converse moves which can describe the total history leading up to s.

[7] *Stronger modal logics of strategies?* The modal μ-calculus is a natural extension of *PDL*, but it lacks explicit programs or strategies, as its formulas merely define properties of states. Is there a version of the μ-calculus that extends *PDL* in defining more transition relations? Say, a simple strategy 'keep playing a' guarantees infinite a-branches for greatest fixed point formulas like $\nu p(\langle a \rangle p)$. [16] looks at richer fragments than *PDL* with explicit programs as solutions to fixed-point equations of special forms, guaranteeing uniform convergence by stage ω.

8 Epistemic Logic of Games with Imperfect Information

The next level of static game structure gives up the presupposition of perfect information. Consider extensive games with *imperfect information*, whose players need not know where they are in a tree. This happens in card games, electronic communication, through bounds on memory or observation. Such games have 'information sets': equivalence classes of relations $\sim_i$ between nodes which players i cannot distinguish. [4] shows how these games model an epistemic modal language including knowledge operators $K_i\varphi$ interpreted in the usual manner as "φ is true at all nodes $\sim_i$-related to the current one".

Example 7 (Partial observation in games). In this imperfect information game, the dotted line indicates player E's uncertainty about her position when her turn comes. Thus, she does not know the move played by player A:[8]



Structures like this are game models of the earlier kind with added epistemic *uncertainty relations* $\sim_i$ for each player. Thus, they interpret a combined dynamic-epistemic language. For instance, after A plays move c in the root, in both middle states, E knows that playing a or b will give her p - as the disjunction $\langle a\rangle p \vee \langle b\rangle p$ is true at both middle states:

$$K_E(\langle a\rangle p \vee \langle b\rangle p)$$

On the other hand, there is no specific move of which E knows at this stage that it will guarantee a p-outcome – and this shows in the truth of the formula

$$\neg K_E\langle a\rangle p \wedge \neg K_E\langle b\rangle p$$

Thus, E knows *de dicto* that she has a strategy which guarantees p, but she does not know, *de re*, of any specific strategy that it guarantees p. Such finer distinctions are typical for a modal language with both actions and knowledge for agents.[9]

We can analyze imperfect information games studying properties of players by modal frame correspondences. An example is the analysis of Perfect Recall for a player i:

[8] Maybe A put his move in an envelope, or E was otherwise prevented from observing.

[9] You may know that the ideal partner for you is around on the streets, but tragically, you might never convert this $K\exists$ combination into $\exists K$ knowledge that some particular person is right for you.

Fact 15. *The axiom $K_i[a]\varphi \rightarrow [a]K_i\varphi$ holds for player i w.r.t. any proposition φ iff M satisfies Confluence:* $\forall xyz : ((xR_a y \wedge y \sim_i z) \rightarrow \exists u : ((x \sim_i u \wedge uR_a z)$.

Similar frame analyses work for memory bounds, and observational powers. For instance, agents satisfy 'No Miracles' when epistemic uncertainty relations can only disappear by observing subsequent events they can distinguish. The preceding game has Perfect Recall, but it violates No Miracles: E suddenly knows where she is after she played her move.

Uniform Strategies. Another striking aspect of our game is *non-determinacy.* E's playing 'the opposite direction from that of player A' was a strategy guaranteeing outcome p in the matching game with perfect information – but it is unusable now. For, E cannot tell if the condition holds! Game theorists only accept uniform strategies here, prescribing the same move at indistinguishable nodes. But then no player has a winning strategy, with p as 'E wins' (and $\neg p$ as a win for player A). A did not have one to begin with, E loses hers.[10]

As for explicit strategies, we can again use *PDL*-style programs, but with a twist. We need the *'knowledge programs'* of [23], whose only test conditions are knowledge statements. In such programs, actions can only be guarded by conditions that the agent knows to be true or false. It is easy to see that knowledge programs can only define uniform strategies. A converse also holds, modulo some mild assumptions on expressiveness of the game language defining nodes in the game tree [4]:

Fact 16. *On expressive finite games of imperfect information, the uniform strategies are precisely those definable by knowledge programs in epistemic* PDL.

9 From Statics to Dynamics: DEL-Representable Games

Now we make a switch. Our approach so far was *static*, using modal-preferential-epistemic logics to describe properties of fixed games. But it also makes sense to look at *dynamic* scenarios, where games can change. As an intermediate step, we analyze how a static game model might have come about by some dynamic process – the way we see a dormant volcano but can also imagine the tectonic forces that shaped it originally. We provide two illustrations, linking games of imperfect information first to dynamic-epistemic logic *DEL*, and then to epistemic-temporal logics *ETL* [28] (cf. [15] on connections). Our sketch will make most sense to readers already familiar with these logics of short-term and long-term epistemic dynamics.

Imperfect Information Games and Dynamic-Epistemic Logic. Dynamic-epistemic logic describes how uncertainty is created systematically as initial uncertainty in an agent model M combines with effects of partially observed events E to create product models $M \times E$. Which imperfect information games 'make sense'

[10] The game does have probabilistic solutions in random strategies: like Matching Pennies.

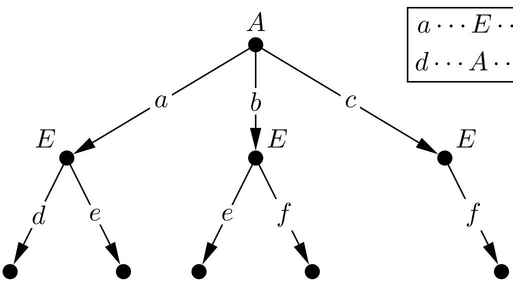
with concrete sequences of update steps – as opposed to being just arbitrary placements of uncertainty links over game forms?

Theorem 17. *An extensive game is isomorphic to a repeated product update model Tree(M, E) for some sequence of epistemic event models E iff it satisfies, for all players: (a) Perfect Recall, (b) No Miracles, and (c) Bisimulation Invariance for the domains of all the move relations.*[11]

Here Perfect Recall is essentially the earlier commutation between moves and uncertainty. We do not prove the Theorem here: cf. [17]. Here is an illustration:

Example 8 (Updates during play: propagating ignorance along a game tree).

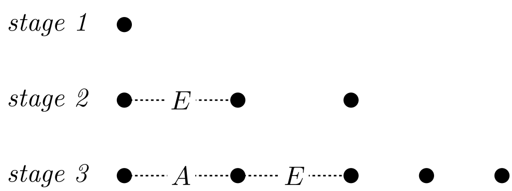
Game tree



Event model

$a \cdots E \cdots b$	c	precondition: $turn_A$
$d \cdots A \cdots e$	f	precondition: $turn_E$

Here are the successive updates that create the right uncertainty links:



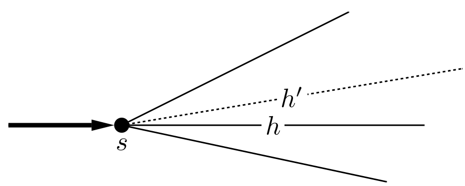
10 Future Uncertainty, Procedural Information, and Branching Temporal Logic

A second logical perspective on games notes that 'imperfect information' has two senses. One is *observation uncertainty*: players may not have seen all events so far, and so they do not know where they are in the game. This is the 'past-oriented' view of *DEL*. But there is also 'future-oriented' *expectation uncertainty*: even in perfect information games players who know where they are may not

[11] This says that two epistemically bisimilar nodes in the game tree make the same moves executable.

know what others, or they themselves, are going to do. The positive side is this. In general, players have some *procedural information* about what is going to happen. Whether viewed negatively or positively, the latter future-oriented kind of knowledge and ignorance need not be reducible to the earlier uncertainty between local nodes. Instead, it naturally suggests current uncertainty between whole future histories, or between players' strategies (i.e., whole ways in which the game might evolve).

Branching Epistemic Temporal Models. The following structure is common to many fields. In tree models for branching time, 'legal histories' h represent possible evolutions of some process. At each stage of the game, players are in a node s on some actual history whose past they know, either completely or partially, but whose future is yet to be fully revealed:



This can be described in an action language with knowledge, belief, and added temporal operators. We first describe games of perfect information (about the past, that is):

- $M, h, s \models F_a\varphi$ iff $s\wedge < a >$ lies on h and $M, h, s\wedge < a > \models \varphi$
- $M, h, s \models P_a\varphi$ iff $s = s'\wedge < a >$, and $M, h, s' \models \varphi$
- $M, h, s \models \Diamond_i\varphi$ iff $M, h', s \models \varphi$ for some h' equal for i to h up to stage s.

Now, as moves are played publicly, players make public observations of them:

Fact 18. *The following valid principle is the* ETL *equivalent of the key* DEL *recursion axiom for public announcement:* $F_a\Diamond\varphi \leftrightarrow (F_a\top \wedge \Diamond F_a\varphi)$.

Trading Future for Current Uncertainty. Again, there is a 'dynamic reconstruction' closer to local *DEL* dynamics. Intuitively, each move by a player is a public announcement that changes the current game model. Here is a folklore observation [6,11] converting 'global' future uncertainty into 'local' present uncertainty:

Fact 19. *Trees with future uncertainty are isomorphic to trees with current uncertainties plus subsequent public announcements.*

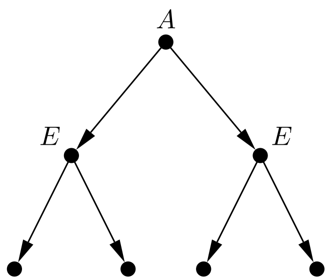
11 Intermezzo: Three Levels of Logical Game Analysis

At this point, it may be useful to distinguish three natural levels at which games have given rise to models for logics. All three come with their own intuitions, both static and dynamic.

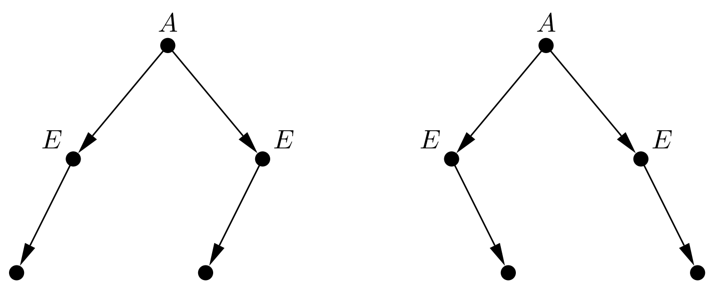
Level One takes *extensive game trees* themselves as models for modal logics, with nodes as worlds, and accessibility relations over these for actions, preferences, and uncertainty. *Level Two* looks at extensive games as *branching tree models*, with nodes and complete histories, supporting richer epistemic-temporal (-preferential) languages. The difference with Level One seems slight in finite games, where histories may be marked by end-points. But the intuitive step seems clear, and also, Level Two does not reduce in this manner when game trees are infinite. But even this is not enough for some purposes!

Consider 'higher' hypotheses about the future, involving procedural information about other players' strategies. I may know that I am playing against either a 'simple automaton', or a 'sophisticated learner'. Modeling this may go beyond epistemic-temporal models:

Example 9 (Strategic uncertainty). In the following simple game, let A know that E will play the same move throughout:



Then all four histories are still possible. But A only considers two *future trees* possible, viz.



In longer games, this difference in modeling can be highly important, because observing only one move by E will tell A exactly what E's strategy will be in the whole game.

To model these richer settings, one needs *Level Three* epistemic *game models.*

Definition 20 (Epistemic game models). Epistemic game models *for an extensive game G are epistemic models $M = (W, \sim_i, V)$ whose worlds are abstract indices including local (factual) information about all nodes in G, plus*

whole strategy profiles for players, i.e., total specifications of everyone's behaviour throughout the game. Players' global information about game structure and procedure is encoded by uncertainty relations $\sim_i$ between worlds of the model.

The above uncertainty between two strategies of my opponent would be naturally encoded in constraints on the set of strategy profiles represented in such a model. And observing some moves of yours in the game telling me which strategy you are actually following then corresponds to dynamic update of the initial model, in the sense of our earlier chapters.

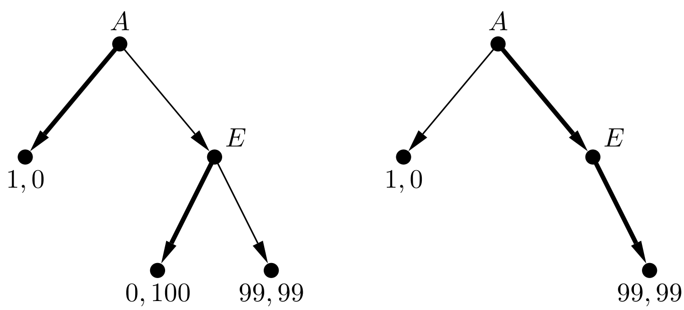
Level-Three models are a natural limit for games and other scenarios of interactive agency. Our policy is always to discuss issues at the simplest model level where they make sense.

12 Game Change: Public Announcements, Promises and Solving Games

Now look at actual transformations that *change games*, and triggers for them.

Promises and Intentions. Following [10], one can break the impasse of a bad Backward Induction solution by changing the game through making *promises.*

Example 10 (Promises and game change). In this earlier game, the 'bad Nash equilibrium' $(1, 0)$ can be avoided by E's promise that she will not go left, by public announcement that some histories will not occur (we may make this binding, e.g., by attaching a huge fine to infractions) – and the new equilibrium $(99, 99)$ results, making both players better off by restricting the freedom of one of them!



But one can also add moves to a game,[12] or give additional information about players' preferences.

Theorem 21. *The modal logic of games plus public announcement is completely axiomatized by the modal game logic chosen, the recursion axioms of PAL for atoms and Booleans, plus the following law for the move modality:*

$$\langle !P\rangle\langle a\rangle\phi \leftrightarrow (P \wedge \langle a\rangle(P \wedge \langle !P\rangle\phi).$$

[12] Yes, in this way, one could code up all such game changes beforehand in one grand initial 'Super Game' – but that would lose all the flavour of understanding what happens in a stepwise manner.

Using PDL again for strategies, this leads to a logic *PDL+PAL* with public announcements $[!P]$. It is easy to show that *PDL* is closed under relativization to definable sub-models, both in its propositional and its program parts, and this underlies the following result:

Theorem 22. *PDL+ PAL is axiomatized by merging their separate laws, while adding the following reduction axiom:*

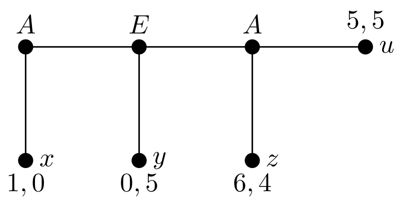
$$[!P]\{\sigma\}\varphi \leftrightarrow (P \to \{\sigma \setminus P\}[!P]\varphi).$$

But of course, we also want to know about versions with epistemic preference languages – and hence there are many further questions following up on these initial observations.

Solving Games by Announcements of Rationality. Another type of public announcement in games iterates various assertions expressing that players are rational, as a sort of 'public reminders'. [9] has this result for extensive games:

Theorem 23. *The Backward Induction solution for extensive games is obtained through repeated announcement of the temporal preferential assertion "no player chooses a move all of whose further histories end worse than all histories after some other available move".*

Proof. This can be proved by a simple induction on finite game trees. The principle will be clear by seeing how the announcement procedure works for a 'Centipede game', with three turns as indicated, branches indicated by name, and pay-offs given for A, E in that order:



Stage 0 of the announcement procedure rules out branch u, Stage 1 then rules out z, while Stage 2 finally rules out y.

This iterated announcement procedure for extensive games ends in largest submodels in which players have common belief of rationality, or other doxastic-epistemic properties.

Alternatives. Of course, a logical language provides many other assertions to be announced, such as history-oriented alternatives, where players steer future actions by reminding themselves of *legitimate rights of other players*, because of 'past favours received'.

The same ideas work in strategic games, using assertions of *Weak Rationality* ("no player chooses a move which she knows to be worse than some other available one") and *Strong Rationality* ("each player chooses a move she thinks may be the best possible one"):

Theorem 24. *The result of iterated announcement of WR is the usual solution concept of Iterated Removal of Strictly Dominated Strategies; and it is definable inside M by means of a formula of a modal μ-calculus with inflationary fixed-points. The same for iterated announcement of SR and game-theoretic Rationalizability.*[13]

13 Belief, Update and Revision in Extensive Games

So far, we studied players' knowledge. We merely indicate how one can also study their equally important beliefs. For a start, one can use Level-One game models with relations of relative plausibility between nodes inside epistemic equivalence classes. Players' beliefs then hold in the most plausible epistemically accessible worlds, and conditional beliefs can be defined as an obvious generalization. But perhaps more vivid is a Level-Two view of branching trees with belief structure. Recall the earlier *ETL* models, and add binary relations $\leq_{I,s}$ of state-dependent *relative plausibility* between histories:

Definition 25 (Absolute and conditional belief). *We set $M, h, s \models \langle B, i\rangle\varphi$ iff $M, h', s \models \varphi$ for some history h' coinciding with h up to stage s and most plausible for i according to the given relation $\leq_{I,s}$. As an extension, $M, h, s \models \langle B, i\rangle^{\psi}\varphi$ iff $M, h', s \models \varphi$ for some history h' most plausible for i according to the given $\leq_{I,s}$ among all histories coinciding with h up to stage s and satisfying $M, h', s \models \psi$.*

Now, belief revision happens as follows. Suppose we are at node s in the game, and move a is played which is publicly observed. At the earlier-mentioned purely epistemic level, this event just eliminates some histories from the current set. But there is now also belief revision, as we move to a new plausibility relation $\leq_{I,s^\wedge a}$ describing the updated beliefs.

Hard Belief Update. First, assume that plausibility relations are not node-dependent, making them global. In that case, we have belief revision under *hard information*, eliminating histories. The new plausibility relation is the old one, restricted to a smaller set of histories. Here is the characteristic recursion law that governs this process. A temporal operator $F_a\phi$ says a is the next event on the current branch, and that ϕ is true immediately after:

Fact 26. *The following temporal principles hold for hard revision along a tree:*

- $F_a\langle B, i\rangle\varphi \leftrightarrow (F_a\top \wedge \langle B, i\rangle(F_a\top, F_a\varphi))$
- $F_a\langle B, i\rangle^{\psi}\varphi \leftrightarrow (F_a\top \wedge \langle B, i\rangle(F_a\psi, F_a\varphi))$[14]

[13] If the iterated assertion A has 'existential-positive' syntactic form (for instance, *SR* does), the relevant definition can be formulated in a standard epistemic μ-calculus.

[14] Similar 'coherence' laws occur in [19], which formalizes games using *AGM* theory.

Soft Update. But belief dynamics is often driven by events of *soft information*, which do not eliminate worlds, but merely rearrange their plausibility ordering [7], as happens in the familiar model-theoretic 'Grove sphere semantics' of belief revision theory. In the above, we already cast Backward Induction in this manner, as a way of creating plausibility relations in a game tree – but beyond such an 'off-line' preprocessing phase of a given game, there can also be dynamic 'on-line' events that might change players' beliefs and expectations in the course of an actual play of the game. With doxastic-temporal models adapted to this setting, we get representation theorems [14] that say which doxastic-temporal models are produced by plausibility update in the style of [2]. Also, [3] provide a striking new dynamic alternative to Aumann-style characterization theorems for Backward Induction.

Further Entanglements: Dynamics of Rationalization. In all our scenarios and logics, knowledge and belief have been entangled notions – and this entanglement even extends to players' preferences [24,26]. But there are many other dynamic scenarios. For instance, [10] discusses *rationalization* of observed behaviour in games, adapting preferences, beliefs, or both, to make observed behaviour rational.

14 Conclusion

We have shown how games naturally involve static and dynamic logics of action, knowledge, belief, and preference. We gave pilot studies rather than grand theory, and we found more open problems than final results. It would be easy to pile up further topics (cf. [12]), pursuing issues of procedural knowledge, soft update and genuine belief revision in games, agent diversity and bounded rationality, infinite games, or connections to explicit automata-theoretic models of agents (as urged by Ram Ramanujam in his 2008 invited lecture at the Workshop on 'Logics of Intelligent Interaction', ESSLLI Hamburg). True. But even at the current level of detail, we hope to have shown that logic and games is an exciting area for research with both formal structure and intuitive appeal.

References

1. Baltag, A., van Ditmarsch, H., Moss, L.: Epistemic Logic and Information Update. In: Adriaans, P., van Benthem, J. (eds.) Handbook of the Philosophy of Information. Elsevier Science Publishers, Amsterdam (to appear, 2008)
2. Baltag, A., Smets, S.: Dynamic Belief Revision over Multi-Agent Plausibility Models. In: Bonanno, G., van der Hoek, W., Wooldridge, M. (eds.) Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision (LOFT 2006), University of Liverpool, pp. 11–24 (2006)
3. Baltag, A., Smets, S., Zvesper, J.: When All is Said and (Not Yet) Done; Dynamic Rationality in Games. In: Proceedings LOFT 2008, University of Amsterdam (2008)

4. van Benthem, J.: Games in Dynamic Epistemic Logic. Bulletin of Economic Research 53(4), 219–248 (2001)
5. van Benthem, J.: Extensive Games as Process Models. Journal of Logic, Language and Information 11, 289–313 (2002)
6. van Benthem, J.: Update and Revision in Games. Lecture notes, ILLC. Amsterdam & Philosophy Stanford (2004)
7. van Benthem, J.: Dynamic Logic of Belief Revision. Journal of Applied Non-Classical Logics 17(2), 129–155 (2007)
8. van Benthem, J.: Logic Games, From Tools to Models of Interaction. In: Gupta, A., Parikh, R., van Benthem, J. (eds.) Logic at the Crossroads, pp. 283–317. Allied Publishers, Mumbai (to appear)
9. van Benthem, J.: Rational Dynamics. International Game Theory Review 9(1), 377–409; Erratum reprint, 9(2), 377–409 (2007)
10. van Benthem, J.: Rationalizations and Promises in Games. In: Philosophical Trends, 'Supplement 2006' on logic, Chinese Academy of Social Sciences, Beijing, pp. 1–6 (2007)
11. van Benthem, J.: The Information in Intuitionistic Logic. Synthese (to appear)
12. van Benthem, J.: Logical Dynamics of Information and Interaction. Institute for Logic, Language and Computation, University of Amsterdam (Book manuscript) (to appear)
13. van Benthem, J.: Logic in Games. Institute for Logic, Language and Computation, University of Amsterdam (Book manuscript) (to appear)
14. van Benthem, J., Dégrémont, C.: Multi-Agent Belief Dynamics: building bridges between dynamic doxastic and doxastic temporal logics. In: Bonanno, G., van der Hoek, W., Loewe, B. (eds.) Proceedings LOFT VIII, University of Amsterdam (2008)
15. van Benthem, J., Gerbrandy, J., Hoshi, T., Pacuit, E.: Merging Frameworks for Interaction. In: Proceedings TARK 2007, University of Namur (2007); To appear in the Journal of Philosophical Logic
16. van Benthem, J., Ikegami, D.: Modal Fixed-Point Logic and Changing Models. In: Avron, A., Dershowitz, N., Rabinovich, A. (eds.) Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of his 85th Birthday. Springer, Berlin (2008)
17. van Benthem, J., Liu, F.: Diversity of Logical Agents in Games. Philosophia Scientiae 8(2), 163–178 (2004)
18. van Benthem, J., van Otterloo, Roy: Preference Logic, Conditionals, and Solution Concepts in Games. In: Lagerlund, H., Lindström, S., Sliwinski, R. (eds.) Modality Matters, University of Uppsala, pp. 61–76 (2006)
19. Bonanno, G.: Temporal Belief Revision Rationalized by Plausibility Orderings. Department of Economics, UC Davis. In: Apt, K., van Rooij, R. (eds.) Games and Interaction, Texts in Logic and Games. Amsterdam University Press
20. Bradfield, J., Stirling, C.: Modal μ-Calculi. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, pp. 721–756. Elsevier, Amsterdam (2006)
21. de Bruin, B.: Explaining Games: on the Logic of Game Theoretic Explanations, Dissertation. ILLC, University of Amsterdam (2004)
22. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic-Epistemic Logic. Cambridge University Press, Cambridge (2007)
23. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. The MIT Press, Cambridge (1995)

24. Girard, P.: Modal Logic for Belief and Preference Change. Dissertation, Department of Philosophy, Stanford University & ILLC Amsterdam (2008)
25. van der Hoek, W., Pauly, M.: Modal Logic for Games and Information. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, pp. 1077–1148. Elsevier, Amsterdam (2006)
26. Liu, F.: Changing for the Better: Preference Dynamics and Preference Diversity. Dissertation DS-2008-02, ILLC, University of Amsterdam (2008)
27. Osborne, M., Rubinstein, A.: A Course in Game Theory. The MIT Press, Cambridge (1994)
28. Parikh, R., Ramanujam, R.: A Knowledge-Based Semantics of Messages. Journal of Logic, Language and Information 12, 453–467 (2003)
29. Venema, Y.: Algebras and Co-Algebras. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, pp. 331–426. Elsevier, Amsterdam (2006)

Machine Checking Proof Theory: An Application of Logic to Logic

Rajeev Goré

Computer Sciences Laboratory
The Australian National University
Canberra
rajeev.gore@anu.edu.au

Abstract. Modern proof-assistants are now mature enough to formalise many aspects of mathematics. I outline some work we have done using the proof-assistant Isabelle to machine-check aspects of proof theory in general, and specifically the proof theory of provability logic GL.

1 Motivation

Proof theory, broadly construed, is the study of derivations as first-class objects. Typically, we study a proof-calculus which captures the notion that a particular formula A is deducible from a given finite collection Γ of assumption formulae in some given logic L: usually written as $\Gamma \vdash_L A$. Typical such calculi are Gentzen's Sequent Calculi, Natural Deduction Calculi or Hilbert Calculi.

But proof theory is error prone. There are numerous examples of published "proofs" in proof theory which have turned out to be incorrect at a later date. These errors often lie undiscovered for years, usually until some diligent Phd student actually tries to work through the proofs in detail and discovers a bug. I give a concrete example later.

Part of the problem is that conferences and journals typically enforce page limits, so that authors are forced to elide full details. Another cause is that proof-theoretical proofs typically contain many similar cases, and humans are notoriously bad at carrying out repetitive tasks with precision. Thus authors often resort to words like "the other cases are similar". But sometimes the errors are very subtle, and are not just a matter of routine checking.

Proof-theoretic proofs often proceed by induction since derivations are usually structured objects like lists, trees or graphs.

Proof assistants are computer programs which allow a user to encode and check proofs written using a special syntax and interface. They have a long history going back to the early 1970s, are usually based upon an encoding of higher-order logic into some extension of Church's typed λ-calculus, and are now an exciting and mature area of research. Indeed, there is now a strong movement to "formalise mathematics" using computers as exemplified by Tom Hales' project to formally verify his "proof" of the Kepler Conjecture `http://code.google.com/p/flyspeck/`.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 23–35, 2009.

Most modern proof-assistants allow us to define infinite sets using inductive definitions. Most also automatically generate powerful induction principles for proving arbitrary properties of such sets.

Given that proof-theory is error-prone and that it typically utilises proofs by induction, can we use modern proof-assistants to help us machine-check proof theory?

An Oath: I have only a limited amount of time and I need to cover a lot of background material. I also want to show you some actual code that we have developed, but I wish to simplify it to hide unimportant details. So here is an oath: I will tell the truth, I may not tell the whole truth, but I won't lie. So complain immediately if you see something blatantly incorrect!

2 Proof Theory: Purely Syntactic Calculi for *L*-Deduction

To begin with the basics, I just want to talk briefly about the proof-calculi we typically study.

We typically work with judgements of the form $\Gamma \vdash_L \Delta$ where Γ and Δ are "collections" of formulae. I deliberately want to leave vague the exact definition of "collection" for now: think of it as some sort of data-structure for storing information.

From these judgements, we usually define rules, and form a calculus by assembling a finite collection of such rules.

A rule typically has a rule name, a (finite) number of premises, a side-condition and a conclusion as shown below:

$$\text{RuleName}\ \frac{\Gamma_1 \vdash_L \Delta_1 \qquad \cdots \qquad \Gamma_n \vdash_L \Delta_n}{\Gamma_0 \vdash_L \Delta_0}\ \text{Condition}$$

We read the rules top-down as statements of the form "if premises hold then conclusion holds", again deliberately using the imprecision of "holds" rather than something more exact.

A derivation of the judgement $\Gamma \vdash_L \Delta$ is typically a finite tree of judgements with root $\Gamma \vdash_L \Delta$ where parents are obtained from children by "applying a rule". From now on, I will usually omit L to reduce clutter.

Figure 1 shows some typical rules from the literature:

Gentzen's LK: in some formulations uses rules built from multisets, but it can also be easily turned into a calculus which uses sets. LK has a particularly pleasing property in that in all its rules, the components of the premises like A and B are subformulae of the components of the conclusion like $A \to B$;

Gentzen's LJ: uses sequents built from multisets, with an added condition that the right hand side must consist of at most one formula. The particular formulation shown also carries its principal formula $A \to B$ from the conclusion into one of its premises, which can be used to show that the contraction rule is redundant;

Calculus	Example Rule	Collection
LK	$(\to L)\ \dfrac{\Gamma, B \vdash \Delta \qquad \Gamma \vdash A, \Delta}{\Gamma, A \to B \vdash \Delta}$	sets of formulae
LJ	$(\to L)\ \dfrac{\Gamma, A \to B \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \to B \vdash C}$	multisets + SOR
ND	$(\{.\}.I)\ \dfrac{\Gamma \vdash K \qquad \Gamma, K \vdash M}{\Gamma \vdash \{M\}_K}$	multisets + SOR
NL	$(\backslash L)\ \dfrac{\Delta \vdash A \qquad \Gamma[B] \vdash C}{\Gamma[(\Delta, A \setminus B)] \vdash C}$	trees with holes
DL	$(\to L)\ \dfrac{X \vdash A \qquad B \vdash Y}{A \to B \vdash (*X) \circ Y}$	complex trees

Fig. 1. Example of Rules of Some Existing Calculi

ND for Security Protocols: Natural deduction calculi have been used to reason about security protocols. This rule captures the idea that the ability to decode the key K and the ability to decode a message M using that key, allows us to decode M even when it is encrypted with key K;

NL: The non-associative Lambek calculus uses sequents built from trees containing "holes". The rule allows us to replace the formula B in a hole inside the tree Γ, with a more complex subtree built from the tree Δ and the formula $A \setminus B$;

Display Logic: Belnap's display logic uses sequents built from complex structural connectives like $*$ and $\circ$ so that its sequents are akin to complex trees.

Thus there are many different notions of "sequent". Our hope is to encode the proofs about such sequents in a modern proof-assistant, specifically Isabelle.

3 Applying a Rule: Example Derivation in Gentzen's LK

In almost all cases, we build derivations in a top-down manner, starting from leaves which are instances of $\Gamma, p \vdash p, \Delta$ by "applying" a rule. Rule application typically proceeds using pattern-matching as exemplified by the following derivation from Gentzen's LK.

Example 1. Here, Γ, A means "Γ multiset-union A".

$$(\wedge \vdash)\ \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \qquad\qquad (\to\vdash)\ \frac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}$$

$$\dfrac{\dfrac{p_0 \vdash p_0, q_0 \qquad p_0, q_0 \vdash q_0}{p_0, (p_0 \to q_0) \vdash q_0}\ (\to\vdash)}{p_0 \wedge (p_0 \to q_0) \vdash q_0}\ (\wedge \vdash)$$

The first rule instance ($\rightarrow\vdash$) utilises the substitutions $\Gamma := \{p_0\}$, $A := p_0$, $B := q_0$ and $\Delta := \{q_0\}$.

The second rule instance ($\wedge \vdash$) utilises the substitutions $\Gamma := \emptyset$, $A := p_0$, $B := p_0 \rightarrow q_0$ and $\Delta := \{q_0\}$.

The example also illustrates the use of sequent calculi as "backward chaining" decision procedures where we can find a derivation starting from $p_0 \wedge (p_0 \rightarrow q_0) \vdash q_0$ and applying the rules in a systematic way "backwards" towards the leaves.

For propositional LK, decidability follows almost immediately by observing that in every such "backward" rule application, at least one formula disappears from the sequents, and is replaced by strictly smaller formulae only. Thus, every branch of rule applications must terminate.

More generally, we typically find some numerical measure which strictly decreases when "reducing" a conclusions to its premises. Indeed, the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, to be held next in Oslo, is dedicated to research aimed at automating reasoning in various non-classical logics using this technique. Some of us have even made a career out of this endeavour!

Notice that the structure of "collections" is significant. For example, the structural rule of contraction shown below at left using multisets is well-defined, with a typical instance shown below at right:

$$(\mathrm{Ctr})\ \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \qquad\qquad \frac{p_0, p_0 \vdash q_0}{p_0 \vdash q_0}\ (\mathrm{Ctr})$$

Similarly, the following contraction lemma is well-defined:

If $\Gamma, A, A \vdash \Delta$ is derivable then so is $\Gamma, A \vdash \Delta$.

But neither makes sense in a setting where sequents are built from sets since the rule instance shown below at left collapses to identity.

$$\frac{p_0, p_0 \vdash q_0}{p_0 \vdash q_0}\ (\mathrm{Ctr}) \qquad\qquad \frac{\{p_0\} \vdash \{q_0\}}{\{p_0\} \vdash \{q_0\}}\ \text{identity}$$

Similarly, the contraction lemma is meaningless since $\Gamma \cup \{A\} \cup \{A\} \vdash \Delta$ is the same as $\Gamma \cup \{A\} \vdash \Delta$.

Although automated reasoning is an important application of sequent calculi, most uses of proof theory are meta-theoretic. For example, proof theory is typically used to answer questions like the following:

Consistency: $\emptyset \vdash_L A$ and $\emptyset \vdash_L \neg A$ are not both derivable;
Disjunction Property: If $\emptyset \vdash_{Int} A \vee B$ then $\emptyset \vdash_{Int} A$ or $\emptyset \vdash_{Int} B$;
Craig Interpolation: If $\Gamma \vdash_L \Delta$ holds then so do $\Gamma \vdash_L A$ and $A \vdash_L \Delta$ for some formula A with $\mathrm{Vars}(A) \subseteq \mathrm{Vars}(\Gamma) \cap \mathrm{Vars}(\Delta)$;
Normal Forms: Is there a (unique) normal form for derivations ?
Curry-Howard: Do normal derivations correspond to well-typed terms of some λ-calculus ?

Equality: When are two derivations of $\Gamma \vdash_L A$ equivalent ?
Relative Strengths: Every derivation in $\vdash_1$ can be simulated polynomially by a derivation in $\vdash_2$

The methods used usually involve reasoning **about** derivations rather than finding derivations, as exemplified by the following lemmas:

Identity: The judgement $A \vdash A$ is derivable for all A.
Monotonicity: If $\Gamma \vdash \Delta$ is derivable then so is $\Gamma, \Sigma \vdash \Delta$.
Exchange: If $\Gamma, A, B \vdash \Delta$ is derivable then so is $\Gamma, B, A \vdash \Delta$.
Contraction: If $\Gamma, A, A \vdash \Delta$ is derivable then so is $\Gamma, A \vdash \Delta$.
Inversion: If the conclusion of a rule instance is derivable then so are the corresponding premise instances.
Cut-elimination/-admissibility: If $\Gamma \vdash A, \Delta$ is (cut-free) derivable and $\Gamma, A \vdash \Delta$ is (cut-free) derivable then so is $\Gamma \vdash \Delta$, where the cut rule is:

$$(\text{cut})\ \frac{\Gamma \vdash A, \Delta \qquad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

Weak/Strong Normalisation: Algorithm to transform a derivation into a normal form by eliminating topmost/nested cuts ?
Cost: How much bigger is the transformed derivation?

4 Proof Theory Is Error-Prone: Provability Logic GL

To illustrate the fact that proof-theory is error-prone, I would like to describe the history of the cut-elimination theorem for a propositional modal logic called GL, after Gödel-Löb.

The logic GL has an Hilbert axiomatisation which extends the standard axiomatisation for modal logic K by adding Löb's axiom $\Box(\Box A \to A) \to \Box A$. It rose to prominence when Solovay showed that $\Box A$ could be interpreted as "A is provable in Peano Arithmetic" [7]. An initial proof-theoretic account was given by Leivant in 1981 when he "proved" cut-elimination for a set-based sequent calculus for GL [3]. But Valentini in 1983 found a simple counter-example and gave a new cut-elimination proof [8]. The issue seemed to have been settled, but in 2001, Moen [5] claimed that Valentini's transformations don't terminate if the sequents $\Gamma \vdash \Delta$ are based on multisets. There is of course no *a priori* reason why a proof based on sets should not carry over with some modification to a proof based on multisets, so this set the cat amongst the pigeons.

In response, Negri [6] in 2005 gave a new cut-elimination proof using sequents built from labelled formulae $w : A$, which captures that the traditional formula A is true at the possible world w. But this is not satisfactory as it brings the underlying (Kripke) semantics of modal logic into the proof theory. Mints in 2005 announced a new proof using traditional methods [4].

But the question of Moen versus Valentini remained unresolved. Finally, Goré and Ramanayake [2] in 2008 showed that Moen is incorrect, and that Valentini's proof using multisets is mostly okay.

Many such examples exist in the literature.

5 Interactive Proof Assistants

Interactive proof-assistants are now a mature technology for "formalising mathematics". They come in many different flavours as indicated by the names of some of the most popular ones `Mizar, HOL, Coq, LEGO, NuPrl, NqThm, Isabelle,` λ`-Prolog, HOL-Lite, LF, ELF, Twelf` $\cdots$ with apologies to those whose favourite proof-assistant I have omitted.

Most of the modern proof-assistants are implemented using a modern functional programming language like ML, which was specifically designed for the implementation of such proof-assistants.

The lowest levels typically implement a typed lambda-calculus with hooks provided to allow the encoding of further logical notions like equality of terms on top of this base implementation. The base implementation is usually very small, comprising of a few hundred lines of code, so that this code can be scrutinised by experts to ensure its correctness.

Almost all aspects of proof-checking eventually compile down to a type-checking problem using this small core, so that trust rests on strong typing and a well-scrutinised small core of (ML) code.

Most proof-assistants also allow the user to create a proof-transcript which can be cross-checked using other proof-assistants to guarantee correctness.

I don't want to go into details, but one type of proof-assistant, called a logical framework, allows the user to manage a proof using the "backward chaining" idea which we saw in use earlier to find derivations using sequent calculi.

Figure 2 shows how these logical frameworks typically work. Thus given some goal β and an inference step which claims that α is implied by β_1 up to β_n, we pattern-match α and β to find their most general unifier θ, and then reduce the original goal β to the n subgoals $\beta_1\theta \cdots \beta_n\theta$.

The pattern matching required is usually (associative-commutative) higher order unification.

The important point is that the logical framework keeps track of sub-goals and the current proof state.

The syntax of the "basic propositions" like α, β is defined via an "object logic", which is a parameter. Different "object logics" can be invoked using the same logical-framework for the task at hand.

The logical properties of ";" like associativity or commutativity, and properties of the "$\Longrightarrow$" like classicality or linearity are determined by the "meta-logic", which is usually fixed for the logical framework in question.

$$
\begin{array}{ll}
[\beta_1 \ ; \ \beta_2 \ ; \ \cdots \ ; \ \beta_n] \Longrightarrow \alpha & \qquad\qquad\qquad \beta \\[2em]
\theta = match(\beta, \alpha) & \qquad \beta_1\theta \ ; \ \beta_2\theta \ ; \ \cdots \ ; \ \beta_n\theta
\end{array}
$$

Fig. 2. Backward Chaining in Logical Frameworks

For example, the meta-logic of Isabelle is higher-order intuitionistic logic. Higher order simply means that functions can accept other functions as arguments and can produce functions as results.

6 Isabelle's LK Object Logic: A Shallow Embedding of Sequent Calculus

We begin with what is called a "shallow embedding" of sequents. The meaning of this term will become apparent as we proceed.

The "propositions" of Isabelle's sequent object logic are sequents built from sequences of formulae as defined in the grammar below:

$$\begin{array}{ll} prop & = sequence \ \texttt{|-} \ sequence \\ sequence & = elem \ (, elem)^* \mid empty \\ elem & = \$id \mid \$var \mid formula \\ formula & = \sim formula \mid formula \ \& \ formula \mid \cdots \end{array}$$

Thus sequents are built from "collections" which are sequences of formulae. A sequent rule built from premise sequents $\beta_1, \cdots, \beta_n$ with conclusion sequent α is encoded directly as the meta-logical expression:

$$[\beta_1 \ ; \cdots \ ; \ \beta_n] \Longrightarrow \alpha$$

Example 2. For example, the (cut) rule shown below is encoded as the meta-logical expression shown below it:

$$(\text{cut}) \ \frac{\Gamma \vdash \Delta, P \qquad\qquad \Gamma, P \vdash \Delta}{\Gamma \vdash \Delta}$$

```
[| $G |- $D,P  ;  $G,P |- $D |] ==> $G |- $D
```

Thus we encode the horizontal bar separating the premises from the conclusion directly using the meta-logical implication $\Longrightarrow$.

The advantage is that we can immediately create and check derivations using the proof assistant to manage the backward chaining involved. That is, we use the proof-assistant to find derivations by applying the rules in a backward way. There is thus a perfect match between the backward chaining involved in finding derivations and the backward chaining involved in the subgoaling provided by the proof-assistant.

The disadvantage is that there is no explicit encoding of a derivation. The derivation is kept implicitly by the proof-assistant and we cannot manipulate its structure. Nor is it possible to encode statements like the identity lemma: the sequent $A \vdash A$ is derivable for all formulae A. It is possible to show that particular instances of this sequent like $P\&Q \vdash P\&Q$ are derivable, but we cannot actually encode the inductive nature of the proof which would require us to show that it held for A being atomic, and that an inductive step would

take us from the case for formulae of length n to formulae of length $n + 1$. In particular, there is no way to state the final step of the induction which allows us to state that the lemma holds for all finite formulae.

7 A Deeper Embedding: Change Object Logic

Recall that the meta-logic provides us with a method for backward chaining via expressions of the form:

$$\beta_1 \ ; \cdots \ ; \ \beta_n \Longrightarrow \alpha$$

The usual method for obtaining the power for reasoning about sequent derivations is to use the full power of higher-order classical logic (HOL) to build the basic propositions β_i.

Isabelle's incarnation of HOL provides the usual connectives of logic like conjunction, disjunction, implication, negation and the higher order quantifiers. But it also provides many powerful facilities allowing us to define new types, define functions which accept and return other functions as arguments, and even define infinite sets using inductive definitions.

For example, the following HOL expressions capture the usual inductive definition of the natural numbers by encoding the facts that "zero is a natural number, and if n is a natural number then so is its successor $s(n)$":

$$\begin{array}{l} \mathtt{0} \in \mathtt{nat} \\ \mathtt{n} \in \mathtt{nat} \Longrightarrow \mathtt{s(n)} \in \mathtt{nat} \end{array}$$

Most proof-assistants will automatically generate an induction principle from a given inductive definition. For example, Isabelle will automatically generate the usual induction principle which states that we can prove a property P holds of all natural numbers if we can show that $P(0)$ holds and we can show that $P(n)$ implies $P(s(n))$. An implicit assumption which facilitates such induction principles is that the inductive definitions are the only way to construct its members. Thus, if n is a natural number, then it is either 0, or is of the form $s(m)$ for some natural number m.

To encode sequent calculus into HOL we first encode the grammar for recognising formulae as below:

```
datatype fml = FC string (fml list)    (* fml connective *)
             | FV string               (* fml variable   *)
             | PP string               (* prim prop      *)
```

There are three type constructors `FC`, `FV` and `PP` which encode formula connectives, formula variables, and atomic formulae (primitive propositions). Each of them takes one string argument which is simply the string we want to use for that construction. The formula connective constructor also accepts a list of formulae, which constitute its subformulae.

For example, `FC "&" [FV "A", PP "q"]` encodes A & q. Since we want to encode modal provability logic GL, we require only the classical connectives, plus two unary modalities `FC "Box" [.]` for $\Box$. and `FC "Dia" [.]` for $\Diamond$..

Isabelle's HOL allows us to form sets and multisets of objects of an arbitrary type, so the HOL expressions `fml set` and `fml multiset` capture the types of formula sets and formula multisets.

Using these types we can build a sequent type using an infix constructor $\vdash$ via:

$$\texttt{datatype seq} \ = \ \texttt{fml multiset} \ \vdash \ \texttt{fml multiset}$$

Isabelle's HOL allows us to form lists of objects of an arbitrary but fixed type, so we define the type of a rule as a pair with the first component being a list of sequent premises and the second component being the conclusion sequent:

$$\texttt{datatype inf} \ = \ (\texttt{seq list} \ , \ \texttt{seq})$$

Finally, we use the HOL type declaration `rli :: inf set` to declare that `rli` is a set of inferences, each a pair of the form `(seq list , seq)`, and inductively define the set `rli` by giving a finite collection of rule instances which belong to this set. For example, the traditional rule $(\& \vdash)$ for introducing a conjunction into the left hand side of a sequent as shown below is given by the encoding below it:

$$(\vdash \&)\ \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A\&B, \Delta}$$

$$(\ [\ \texttt{G} \vdash \{\texttt{A}\} + \texttt{D}\ ,\ \texttt{G} \vdash \{\texttt{B}\} + \texttt{D}\ ],\ \ \texttt{G} \vdash \{\texttt{A\&B}\} + \texttt{D}\ \ ) \in \texttt{rli}$$

The encoding uses HOL's notation "+" for multiset union, and a slightly inaccurate description of encoding singleton multisets as `{A}`. Thus each element of `rli` is a pair whose first component is a list of its premises, and whose second component is its conclusion.

We are now in a position to encode the set `derrec` of "recursively derivable sequents" given an initial set `pms` of premise sequents and an initial set `rli` of inference rules. The set `derrec` is defined inductively as shown below:

```
1    derrec :: (seq list, seq) set ⇒ seq set ⇒ seq set
2    c ∈ pms ⟹ c ∈ derrec rli pms
3    [ (ps, c) ∈ rli ;
4       ∀ p. p ∈ (set ps) ⟹ p ∈ derrec rli pms ]
5    ⟹ c ∈ derrec rli pms
```

The explanation is as below:

1: A type declaration which tells the proof-assistant that `derrec` accepts a set of inference rules and a set of sequents, and produces a set of sequents;

2: The base case of the inductive definition of `derrec` captures that "each premise is itself (vacuously) derivable from the premises using the rules". Note that there is an implicit outermost universal quantifier which is not shown explicitly, but which binds free variables like `c`, `ps`, `rli`, `pms`.

3: The first conjunct of an inductive clause stating that ps/c is a rule instance;

4: The second conjunct of the inductive clause which captures that "each premise p in the set obtained from sequent list ps is derivable from the premises `pms` using the rules `rli`". Here we use a function `set` to convert a list into the set of its members;
5: The "then" part of the inductive clause which concludes that sequent "c is derivable from `pms` using `rli`".

8 Inductive Proofs via Automated Inductive Principles

Induction principles are generated automatically by Isabelle from the inductive definition of `derrec`. A heavily simplified version for proving an arbitrary property `P` is shown below:

1 $\quad \forall \mathtt{x}.\forall \mathtt{P}.$
2 $\quad [\ \ \mathtt{x} \in \mathtt{derrec\ rli\ pms}\ ;$
3 $\quad\ \ \forall \mathtt{c}.\mathtt{c} \in\ \mathtt{pms} \Longrightarrow\ \mathtt{P(c)}\ ;$
4 $\quad\ \ \forall \mathtt{c}.\forall \mathtt{ps}.[\ \ (\mathtt{ps}, \mathtt{c}) \in\ \mathtt{rli}\ \ ;\ \forall \mathtt{y} \in (\mathtt{set\ ps}).\ \mathtt{P(y)} \Longrightarrow\ \mathtt{P(c)}\ ]$
5 $\quad]\ \Longrightarrow\ \mathtt{P(x)}$

An explanation is:

1: for all sequents x and all properties P
2 : if x is derivable from premises `pms` using rules `rli`, and
3 : P holds for every premise c in `pms`, and
4 : for each rule, if P of its premises implies P of its conclusion,
5 : then P holds of x

If you look closely, you will see that this is an induction principle which we use often in proof-theory: prove that some property holds of the leaves of a derivation, and prove that the property is preserved from the premises to the conclusion of each rule. For example, consider the standard translation from sequents of LK to formulae given by $\tau(A_1, \cdots, A_n \vdash B_1, \cdots, B_m) = A_1 \wedge \cdots \wedge A_n \rightarrow B_1 \vee \cdots \vee B_m$. We typically use this translation to argue that all derivable sequents are valid in the semantics of first-order logic. The proof proceeds by showing that the translation of the leaves of a derivation are all valid, and showing that if the translations of the premises are valid then the translations of the conclusion are valid, for every rule.

Using these inductive principles we proved the following lemma about derivability using Isabelle, where the question marks indicate free-variables which are implicitly universally quantified:

Lemma 1

$$?\mathtt{ps} \subseteq \mathtt{derrec}\ ?\mathtt{rli}\ ?\mathtt{pms}\ ;\ ?\mathtt{c} \in \mathtt{derrec}\ ?\mathtt{rli}\ ?\mathtt{ps}\ \Longrightarrow ?\mathtt{c} \in \mathtt{derrec}\ ?\mathtt{rli}\ ?\mathtt{pms}$$

If each premise in ps is derivable from premises `pms` *using rules* `rli`*, and c is derivable from ps using* `rli`*, then c is derivable from* `pms` *using* `rli`*.*

9 An Even Deeper Embedding: Derivation Trees as Objects

The main advantage of the method outlined in the previous section was that there was no concrete representation of a derivation. That is, we relied on the proof-assistant to perform pattern-matching and rule instantiations in an appropriate way, so that all we needed was to capture the idea that derivations began with premises and ended with a single sequent.

If we are to reason about cut-elimination, then we are required to perform transformations on explicit derivations. We therefore need a representation of such trees inside our encoding.

In previous work [1], we described such an encoding using the following datatype:

```
datatype seq dertree = Der seq (seq dertree list)
                     | Unf seq
```

The declaration states that a derivation tree can either be an `Unf`inished leaf sequent built using the constructor `Unf`, or it can be a pair consisting of a conclusion sequent and a list of sub-derivation-trees bound together using the constructor `Der`.

In that work, we described how we maintained substitutions as lists of pairs of the form (x, t) representing the substitution $x := t$. We also described how we manipulated substitutions and instantiation directly to obtain rule instances.

We required such low-level aspects to be made explicit so that we could reason about display logic which required us to check conditions on rules like “a rule is closed under substitution of arbitrary structures for variables”.

Our use of `dertee` can be seen as an even deeper embedding of proof-theory into Isabelle/HOL since we utilise the proof-assistant only to maintain the current and further goals.

Omitting details now, suppose we define `valid rli dt` to hold when derivation tree `dt` uses rules from `rli` only and has no `Unf`inished leaves. We proved:

Lemma 2

$$\texttt{valid}\ ?\texttt{rli}\ ?\texttt{dt} \implies (\texttt{conclDT}\ ?\texttt{dt}) \in \texttt{derrec}\ ?\texttt{rls}\ \{\}$$

If derivation tree dt is valid wrt the rules `rli` *then its conclusion is derivable from the empty set of premises using* `rli`*.*

Lemma 3

$$?\texttt{c} \in \texttt{derrec}\ ?\texttt{rli}\ \{\} \implies \texttt{EX dt.}\ \texttt{valid}\ ?\texttt{rli}\ \texttt{dt}\ \&\ \texttt{conclDT}\ \texttt{dt}\ =\ ?\texttt{c}$$

If the sequent c is derivable from the empty set of premises using rules `rli` *then there exists a derivation tree dt which is valid wrt* `rli` *and whose conclusion is exactly c.*

Thus we now know that our "deep embedding" of derivability using `derrec` can be faithfully captured using the "even deeper" embedding using explicit derivation trees. Indeed, the lemmas allow us to move freely between the two embeddings at will to omit or include details as required by the lemma we wish to prove.

10 Mix Admissibility for Provability Logic

We finally come to the crux of our work. Below is the traditional formulation of the mix-rule for sequents built from multisets where Π_A is formed from Π by deleting all occurrences of A:

$$(\text{mix})\ \frac{\Gamma \vdash \Delta \qquad \Pi \vdash \Sigma}{\Gamma, \Pi_A \vdash \Delta_A, \Sigma}\ A \in \Delta\ \ \&\ \ A \in \Pi$$

The rule can be expressed as a lemma rather than a rule using the embeddings we have developed so far as shown below where we now explicitly use the name `glss` for the fixed but inductively defined set of rule instances for provability logic GL:

$$(\texttt{?G} \ \vdash\ \texttt{?D}) \in \texttt{derrec glss}\ \{\}\ \ ;\ \ (\texttt{?P} \ \vdash\ \texttt{?S}) \in \texttt{derrec glss}\ \{\}$$
$$\Longrightarrow$$
$$((\texttt{?G} \ +\ (\texttt{ms\_delete}\ \{\texttt{?A}\}\ \texttt{?P}) \ \vdash\ (\texttt{ms\_delete}\ \{\texttt{?A}\}\ \texttt{?D})\ +\ \texttt{?S})) \in \texttt{derrec glss}\ \{\}$$

Here we defined a function `ms_delete` which deletes all occurrences of its first argument from its second argument. Our main result, which we intend to report in a proper paper, is that this lemma can be proved using our embeddings and Isabelle.

11 Objections and Impediments

A frequent objection to the idea of machine-checking anything is that the errors could also have been found by a good Phd student working with pencil and paper. But even diligent Phd students are apt to fall for errors which lie within sentences marked by "clearly ..." or the "other cases are similar". The beauty of proof-assistants lies in their absolutely pedantic insistence that nothing is proved until it passes through the type-checking procedure of the proof-assistant.

Another objection is that this is not research but is just high level programming since you have to have the proof first. To some extent this is true since the current prototypical example is usually the verification of a given proof from a paper or a book. But many researchers now build the proof interactively. Indeed, better user-interfaces make this very possible.

The main impediment in my opinion is the sheer effort required to become familiar with proof-assistants before productive work can be started. It takes at least three months of full-time work to learn how to use an interactive proof assistant well. But as I hope I have shown you, it is worth it!

Acknowledgements. Most of the work reported here has been done in collaboration with Jeremy E Dawson. He is a perfect example of a serious mathematician who checks all of his proofs in Isabelle ... just to be sure.

References

1. Dawson, J.E., Goré, R.: Formalised Cut Admissibility for Display Logic. In: Carreño, V.A., Muñoz, C.A., Tahar, S. (eds.) TPHOLs 2002. LNCS, vol. 2410, pp. 131–147. Springer, Heidelberg (2002)
2. Goré, R., Ramanayake, R.: Valentini's Cut Elimination for Provability Logic Resolved. In: Advances in Modal Logic, vol. 7, pp. 67–86. College Publications, London (2008)
3. Leivant, D.: On the Proof Theory of the Modal Logic for Arithmetic Provability. Journal of Symbolic Logic 46, 531–538 (1981)
4. Mints, G.: Cut elimination for provability logic. Collegium Logicum (2005)
5. Moen, A.: The proposed algorithms for eliminating cuts in the provability calculus *GLS* do not terminate. In: NWPT 2001, Norwegian Computing Center, December 10 (2001), `http://publ.nr.no/3411`
6. Negri, S.: Proof Analysis in Modal Logic. Journal of Philosophical Logic 34, 507–544 (2005)
7. Solovay, R.M.: Provability Interpretations of Modal Logic. Israel Journal of Mathematics 25, 287–304 (1976)
8. Valentini, S.: The Modal Logic of Provability: Cut-elimination. Journal of Philosophical Logic 12, 471–476 (1983)

Some Second Order Set Theory

Joel David Hamkins⋆

The City University of New York
The Graduate Center of CUNY, Mathematics
365 Fifth Avenue, New York, NY 10016
and
The College of Staten Island of CUNY, Mathematics
2800 Victory Boulevard, Staten Island, NY 10314
jhamkins@gc.cuny.edu
http://jdh.hamkins.org

Abstract. This article surveys two recent developments in set theory sharing an essential second-order nature, namely, the modal logic of forcing, oriented upward from the universe of set theory to its forcing extensions; and set-theoretic geology, oriented downward from the universe to the inner models over which it arises by forcing. The research is a mixture of ideas from several parts of logic, including, of course, set theory and forcing, but also modal logic, finite combinatorics and the philosophy of mathematics, for it invites a mathematical engagement with various philosophical views on the nature of mathematical existence.

1 Introduction

I would like in this article to discuss two emerging developments in set theory focusing on second-order features of the set-theoretic universe, and focusing particularly on the relation of the universe of sets in a general context to other more arbitrary models. The first of the these developments, the modal logic of forcing, has an upward-oriented focus, looking upwards from a model of set theory to its extensions and investigating the relationship of the model to these extensions and their subsequent relation to further extensions. The second development, set-theoretic geology, has a downward-oriented focus, looking from a model of set theory down to the models of which it is an extension, and investigating the resulting structure of this collection of models. These two perspectives are unified by and find motivation in a multiverse view of set theory, the philosophical view that there are many set-theoretic worlds. Indeed, such a philosophical view has perhaps guided the mathematical research in this area by suggesting what have

⋆ My research has been supported in part by grants from the CUNY Research Foundation and from the National Science Foundation, for which I am very grateful. This article is the preliminary text version of a talk I expect to give for the Third Indian Conference on Logic and its Applications (ICLA), sponsored by the Association for Logic in India and held at the Institute of Mathematical Sciences in Chennai, India Jaunary 7–11, 2009.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 36–50, 2009.

turned out to be very interesting questions and also what have turned out to be productive avenues for research. The work I shall discuss can be taken as the initial footsteps in what I hope will become a thorough mathematical exploration of this philosophical view. This brief survey is intended to introduce the area by describing the principal concepts and preliminary results, mostly adapted from [3], [4] and [1], but with the main proofs only sketched here. I shall call particular attention to the many interesting and fundamental questions that remain open, and I invite researchers to the topic.

2 Looking Upward: The Modal Logic of Forcing

Although many set-theorists affirm the Platonic view that there is just one universe of set theory, nevertheless the most powerful set-theoretic tools developed over the past half century are actually methods of constructing alternative universes. With both the method of forcing and the method of ultrapowers—and these two methods can be viewed as two facets of the single method of Boolean ultrapowers[1]—a set theorist begins with a model of set theory V and constructs another model W by forcing or by ultrapowers (for example, via large cardinal embeddings), making set-theoretic progress by means of a detailed investigation of the often close connection between V and W. And of course set theorists, ever tempted by the transfinite, perform very long iterations of these methods, sometimes intertwining them in combination, to gain even greater understanding and construct additional models of set theory.

Forcing, introduced by Paul Cohen in 1963, is a method for constructing a larger model of set theory extending a given model. Cohen used the method to settle the independence of the Continuum Hypothesis CH from the other axioms of ZFC, by showing that every model of set theory has a forcing extension in which CH fails. In a subsequent explosion of applications, set theorists have constructed an enormous variety of models of set theory, often built to exhibit certain precise, exacting features, and we have come thereby to see the rich diversity of mathematical possibility.

With forcing, one begins with a ground model $V \models \mathrm{ZFC}$ and a partial order or forcing notion $\mathbb{P}$ in V. The forcing extension $V[G]$, a model of ZFC, is built by adjoining an ideal generic element G, a V-generic filter $G \subseteq \mathbb{P}$, in a manner akin to a field extension. In particular, the ground model has *names* for every element of the forcing extension $V[G]$, and every object of $V[G]$ is constructible algebraically from these names in the ground model and the new object G. Much of the power of forcing flows from the surprising degree of access the ground model V has to the objects and the truths of the extension $V[G]$. The overall effect is that the forcing extension $V[G]$ is closely related to the ground model V, but exhibits new truths in a way that can be carefully controlled.

So let us consider a model of set theory V and its relation to all its forcing extensions $V[G]$, considering at first only extensions by set forcing. It seems very natural to introduce the idea that a statement φ in the language of set

[1] This idea is fully explored in the forthcoming [6].

theory is *forceable* or *possible* if there is some forcing extension $V[G]$ in which φ is true. This is, of course, a modal possibility notion, so we will write $\Diamond\varphi$ for the assertion that φ is forceable. The natural dual notion is that φ is *necessary*, written $\Box\varphi$, when φ holds in all forcing extensions $V[G]$. There is, of course, a natural Kripke model lurking here, whose possible worlds are the models of set theory and whose accessibility relation is the relation of a model to its forcing extensions. Many set theorists habitually operate within this Kripke model, even if they would not describe their activities this way, for whenever it is convenient and for whatever purpose they say, "let $G \subseteq \mathbb{P}$ be V-generic," and make the move to the forcing extension $V[G]$. This amounts to traveling about in this Kripke model.

The modal assertions $\Diamond\varphi$ and $\Box\varphi$ are expressible, of course, in the language of set theory.

$$\Diamond\varphi \qquad \iff \qquad \exists\mathbb{P}\,\exists p \in \mathbb{P}\; p \Vdash_{\mathbb{P}} \varphi$$

$$\Box\varphi \qquad \iff \qquad \forall\mathbb{P}\,\forall p \in \mathbb{P}\; p \Vdash_{\mathbb{P}} \varphi$$

The forcing relation $p \Vdash_{\mathbb{P}} \varphi$ means that whenever $G \subseteq \mathbb{P}$ is a V-generic filter and $p \in G$, then the resulting forcing extension $V[G]$ satisfies φ. Two of the most fundamental facts about forcing, central to the entire forcing enterprise, are expressed by the Forcing Lemmas, which assert, first, that every statement φ true in a forcing extension $V[G]$ is forced by some condition $p \in G$, and second, that for φ of fixed complexity, the forcing relation $p \Vdash_{\mathbb{P}} \varphi$ is definable from parameters in the ground model. These lemmas express precisely the sense in which the ground model has access to the truths of the forcing extension. It follows now that both $\Diamond\varphi$ and $\Box\varphi$ are expressible in the language of set theory. And while $\Diamond$ and $\Box$ are therefore eliminable, we nevertheless retain them, for we are interested in what principles these operators must obey.

Many common elementary modal assertions, it is easy to see, are valid under this forcing interpretation. To be precise, let me define that a modal assertion $\varphi(p_0, \ldots, p_n)$, in the language of propositional modal logic, is a *valid principle of forcing* if for any set-theoretic assertions $\psi_0, \ldots, \psi_n$ the corresponding substitution instance $\varphi(\psi_0, \ldots, \psi_n)$ holds. For example, it is easy to see that the following elementary modal assertions are valid principles of forcing.

$$\begin{array}{rl}
\text{K} & \Box(\varphi \implies \psi) \implies (\Box\varphi \implies \Box\psi) \\
\text{Dual} & \Box\neg\varphi \iff \neg\Diamond\varphi \\
\text{S} & \Box\varphi \implies \varphi \\
4 & \Box\varphi \implies \Box\Box\varphi \\
.2 & \Diamond\Box\varphi \implies \Box\Diamond\varphi
\end{array}$$

Since these assertions axiomatize the modal theory known as S4.2, it follows that:

Theorem 1. *Every* S4.2 *modal assertion is a valid principle of forcing.*

The fundamental question is:

Question 2. *What are the valid principles of forcing?*

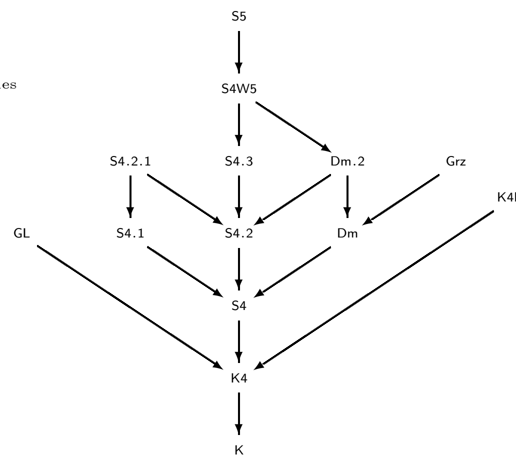
An answer will be provided by Theorem 3. As a tentative first step, let me mention that it is an enjoyable elementary exercise in forcing, which I encourage the reader to undertake, to show that none of the following modal assertions is a valid principle of forcing in every model of set theory.

5	$\Diamond\Box\varphi \implies \varphi$
M	$\Box\Diamond\varphi \implies \Diamond\Box\varphi$
W5	$\Diamond\Box\varphi \implies (\varphi \implies \Box\varphi)$
.3	$\Diamond\varphi \wedge \Diamond\psi \implies (\Diamond(\varphi \wedge \Diamond\psi) \vee \Diamond(\varphi \wedge \psi) \vee \Diamond(\psi \wedge \Diamond\varphi))$
Dm	$\Box(\Box(\varphi \implies \Box\varphi) \implies \varphi) \implies (\Diamond\Box\varphi \implies \varphi)$
Grz	$\Box(\Box(\varphi \implies \Box\varphi) \implies \varphi) \implies \varphi$
Löb	$\Box(\Box\varphi \implies \varphi) \implies \Box\varphi$
H	$\varphi \implies \Box(\Diamond\varphi \implies \varphi)$

As a hint for this exercise, let me mention that several of the assertions above are invalid in every model of set theory, with $\varphi = \mathrm{CH}$ (or its negation) being a counterexample. The others are invalid in L (and other models), with counterexamples built from such assertions as $V \neq L$, $\omega_1^L < \omega_1$, CH, or Boolean combinations of these. The axioms above correspond to a hierarchy of modal theories:

Some Common Modal Theories

S5	=	S4 + 5
S4W5	=	S4 + W5
S4.3	=	S4 + .3
S4.2.1	=	S4 + .2 + M
S4.2	=	S4 + .2
S4.1	=	S4 + M
S4	=	K4 + S
Dm.2	=	S4.2 + Dm
Dm	=	S4 + Dm
Grz	=	K + Grz = S4 + Grz
GL	=	K4 + Löb
K4H	=	K4 + H
K4	=	K + 4
K	=	K + Dual



The forcing interpretation of the modal operators $\Box\varphi$ and $\Diamond\varphi$ was introduced in [3], in connection with the forcing axiom called the Maximality Principle MP, which was fruitfully cast in these modal terms.[2] Specifically, having the concept of a set theoretical assertion φ being forceable or being necessary, we define that φ is *forceably necessary* if it is forceable that φ is necessary, that is, if we can force φ in such a way that it remains true in all further forcing extensions. The Maximality Principle is the scheme asserting that every forceably necessary statement is already true. In our modal notation, this is simply the assertion $\Diamond\Box\varphi \implies \varphi$, which happens to be the modal axiom known as S5. Thus, the

[2] See [10] for an earlier independent account, without the modal interpretation, of a version of MP.

Maximality Principle is simply the assertion that S5 is a valid principle of forcing. Elementary modal reasoning shows that MP is equivalent, as a scheme, to the scheme asserting that every forceably necessary statement is not only true, but also necessary, expressed by $\Diamond\Box\varphi \implies \Box\varphi$. In [3], it was proved that if there is a model of ZFC, then there is a model of ZFC + MP. Although this was a forcing argument, it involved a certain interesting non-forcing hiccup in the choice of ground model, and it is not true that every model of ZFC has a forcing extension that is a model of MP. Indeed, if ZFC is consistent, then there is a model of ZFC having no extension of any kind with the same ordinals that is a model of MP. The original forcing axioms, from Martin's Axiom onwards, have often been cast and were originally conceived (according to my conversations with Tony Martin) as asserting that a lot of forcing has already occurred. The Maximality Principle makes this idea universal, by asserting that any statement that could be forced necessary is already necessary.

But of course, the Maximality Principle does not hold in all models of set theory, so the question remains: What are the valid principles of forcing? The following theorem, the main theorem of [4], provides an answer.

Theorem 3 (Hamkins, Löwe [4]). *If* ZFC *is consistent, then the* ZFC*-provably valid principles of forcing are exactly those in the modal theory* S4.2.

Let me mention a few concepts from the proof. We have already observed above that S4.2 is valid for forcing. The difficult part of the theorem, of course, is to show that there are no other validities. In other words, given S4.2 $\nvdash \varphi$, we must provide set-theoretic assertions ψ_i such that $\varphi(\psi_0, \ldots, \psi_n)$ fails in some model of set theory. To accomplish this, two attractively simple concepts turn out to be key. Specifically, we define that a statement φ of set theory is a *switch* if both φ and $\neg\varphi$ are necessarily possible. Thus, a switch is a statement φ whose truth value can always be turned on or off by further forcing. In contrast, φ is a *button* if φ is (necessarily) possibly necessary. These are the statements that can be forced true in such a way that they remain true in all further forcing extensions. The idea here is that once you push a button, you cannot unpush it. The Maximality Principle, for example, is equivalent to the assertion that every button has already been pushed. Although buttons and switches may appear at first to be very special kinds of statements, it is nevertheless the case in set theory that every statement is either a button, a switch, or the negation of a button. (After all, if you can't always switch φ on and off, then it will either get stuck on or stuck off, and product forcing shows these possibilities to be mutually exclusive.) A family of buttons and switches is *independent*, if the buttons are not yet pushed and (necessarily) each of the buttons and switches can be controlled without affecting the others. Under $V = L$, there is an infinite independent family of buttons and switches, namely, b_n = "ω_n^L is collapsed" and s_m = "GCH holds at $\aleph_{\omega+m}$" (for $n, m > 0$), since the truth of these statements can be controlled independently by forcing.

The proof of Theorem 3 rests in part on a detailed understanding of the modal logic S4.2 and its complete sets of Kripke frames. A *Kripke model* is a collection of propositional *worlds* (essentially a truth table row, assigning propositional

variables to true and false), with an underlying accessibility relation called the *frame*. A statement is possible or necessary at a world, accordingly as it is true in some or all accessible worlds, respectively. Every Kripke model built on a frame that is a directed partial pre-order will satisfy the S4.2 axioms of modal logic, and in fact the finite directed partial pre-orders are *complete* for S4.2 in the sense that the statements true in all Kripke models built on such frames are exactly the statements provable from S4.2. An improved version of this, proved in [4], is that the finite pre-lattices, and even the finite pre-Boolean algebras, are complete for S4.2. The following lemma, a central technical claim of [4], shows that any model of set theory with an independent family of buttons and switches is able to simulate any given Kripke model built on a finite pre-lattice frame.

Lemma 4. *If $W \models$ ZFC has sufficient independent buttons and switches, then for any Kripke model M on a finite pre-lattice frame, any $w \in M$, there is a translation of the propositional variables $p_i \mapsto \psi_i$ to set-theoretic assertions ψ_i, such that for any modal assertion $\varphi(p_1, \ldots, p_n)$:*

$$(M, w) \models \varphi(p_1, \ldots, p_n) \iff W \models \varphi(\psi_1, \ldots, \psi_n).$$

Each ψ_i is a Boolean combination of the buttons and switches.

Consequently, if S4.2 $\nvdash \varphi$, then since we proved that there is a Kripke model M built on a finite pre-lattice frame in which φ fails, it follows that in any model of set theory W having independent buttons and switches, which we proved exist, the corresponding assertion $\varphi(\psi_1, \ldots, \psi_n)$ fails. This exactly shows that φ is not a provably valid principle of forcing, as desired to prove Theorem 3. The proof is effective in the sense that if S4.2 $\nvdash \varphi$, then we are able explicitly to provide a model $W \models$ ZFC and the particular set-theoretic substitution instance $\varphi(\psi_1, \ldots, \psi_n)$ which fails in W.

Although Theorem 3 tells us what are the ZFC-provably valid principles of forcing, it does not tell us that all models of ZFC exhibit only those validities. Indeed, we know that this isn't the case, because we know there are models of the Maximality Principle, for which the modal theory S5 is valid, and this is strictly stronger. So different models of set theory may exhibit different valid principles of forcing. For any $W \models$ ZFC, consider the family Force^W of modal assertions φ that are valid for forcing over W. The proof of Theorem 3 can be adapted to show that

Theorem 5 (Hamkins, Löwe [4]). *If $W \models$ ZFC, then S4.2 $\subseteq \text{Force}^W \subseteq$ S5.*

Furthermore, both of these endpoints occur, and so the theorem is optimal. Specifically, if W is a model of $V = L$, then $\text{Force}^W =$ S4.2, and if W satisfies the Maximality Principle, then $\text{Force}^W =$ S5.

Questions 6

1. *Is there a model of* ZFC *whose valid principles of forcing form a theory other than* S4.2 *or* S5*?*

2. *If φ is valid in W, is it valid in all extensions of W?*
3. *Equivalently, is* Force^W *normal?*
4. *Can a model of* ZFC *have an unpushed button, but not two independent buttons?*

The validity for forcing of many modal axioms can be re-cast in purely set-theoretic terms, in the button-and-switch manner. For example, a model $W \models$ ZFC has no unpushed buttons if and only if $\mathrm{Force}^W = \mathsf{S5}$, and W has independent buttons and switches if and only if $\mathrm{Force}^W = \mathsf{S4.2}$. Moving beyond this, if W has two semi-independent buttons (meaning that the first can be pushed without pushing the second), then W5 invalid in W; If W has two independent buttons, then .3 is invalid in W; If W has an independent button and switch, then Dm is invalid in W; And if W has long volume controls (sequences of buttons, such that each can be pushed without pushing the next and pushing any of them necessarily pushes all earlier buttons—so the volume only gets louder), then $\mathrm{Force}^W \subseteq \mathsf{S4.3}$.

When parameters are allowed into the scheme, large cardinals make a surprising entrance.

Theorem 7. *The following are equiconsistent:*

1. $\mathsf{S5}(\mathbb{R})$ *is valid.*
2. $\mathsf{S4W5}(\mathbb{R})$ *is valid for forcing.*
3. $\mathsf{Dm}(\mathbb{R})$ *is valid for forcing.*
4. *There is a stationary proper class of inaccessible cardinals.*

Theorem 8.

1. *(Welch, Woodin) If* $\mathsf{S5}(\mathbb{R})$ *is valid in all forcing extensions (using the* $\mathbb{R}$ *of the extension), then* $\mathrm{AD}^{L(\mathbb{R})}$.
2. *(Woodin) If* $\mathrm{AD}_{\mathbb{R}} + \Theta$ *is regular, then it is consistent with* ZFC *that* $\mathsf{S5}(\mathbb{R})$ *is valid in all forcing extensions.*

There are many directions for future work in this area. In addition to the questions above, it is natural to restrict the class of forcing to ccc forcing, or proper forcing or any other natural class of forcing.

Questions 9. *What are the valid modal principles of* ccc *forcing? Of proper forcing? Of class forcing? Of arbitrary extensions?*

Class forcing and arbitrary extensions involve the meta-mathematical complication that the corresponding possibility and necessitation operators are no longer first-order expressible. The work on Question 9 has been surprisingly difficult, even for what we expected would be the easier cases, and has led to some interesting, subtle questions in forcing combinatorics. For example, the question of whether there must be switches in the modal logic of collapse forcing (the class of all forcing $\mathrm{Coll}(\omega, \delta)$ to collapse a cardinal δ to ω using finite conditions, and more generally also the Lévy collapse $\mathrm{Coll}(\omega, {<}\delta)$) leads directly to the following question:

Question 10. *Can there be a model of set theory V that is elementarily equivalent to $V[G]$, whenever G is V-generic for the collapse of a cardinal δ to ω?*

Such a model of set theory would be an extreme counterexample in having no switches at all for the class of collapse forcing, and would have valid principles of collapse forcing that are beyond $\mathsf{S5}$, a hard upper bound for the other natural classes of forcing. Mitchell and Welch have given lower bounds with large values of $o(\kappa)$, but for the upper bound, an early suggestion of Mitchell to perform Radin forcing over a model of $o(\kappa) = \kappa^+$ has reportedly not worked out as hoped.

3 Looking Downward: Set-Theoretic Geology

Let me turn now to a second topic, a collection of problems and results we have called set-theoretic geology. Forcing is ordinarily viewed as a method of constructing *outer* as opposed to *inner* models of set theory, for with forcing, as I explained above, one usually begins with a ground model V and builds the forcing extension $V[G]$ by adjoining G and constructing relative to V. Nevertheless, a simple switch in perspective allows us to use forcing to describe inner models as well. The idea is simply to consider forcing from the perspective of the forcing extension rather than the ground model and to look downward from the universe V for how it may have arisen by forcing. Given the set-theoretic universe V, we search for the possible ground models $W \subseteq V$ such that there is a W-generic filter $G \subseteq \mathbb{P} \in W$ such that $V = W[G]$. Such a perspective quickly leads one to look for deeper and deeper grounds, burrowing down to what we call bedrock models and deeper still, to what we call the mantle and the outer core. In this way, one arrives at set-theoretic geology. The topic is introduced in [1], which gives the initial results and numerous open questions, and the material here is adapted from that article.

The topic rests fundamentally on the following theorem, a shockingly recent result, considering the fundamental nature of the question it answers. Laver's proof of this theorem builds on work of mine [2] concerning the approximation and covering properties.

Theorem 11 (Laver [7], independently Woodin [11]). *Every model of set theory $V \models$ ZFC is a definable class in all of its set forcing extensions $V[G]$, using parameters in V.*

This theorem led Jonas Reitz and me to introduce the following hypothesis, which we take to be the beginning of set-theoretic geology.

Definition 12 (Hamkins, Reitz). The Ground Axiom GA is the assertion that the universe is not obtained by nontrivial set forcing over any inner model.

Although this assertion may appear at first to be second order, because of the quantification over ground models, in fact the Ground Axiom is expressible by a first order statement in the language of set theory.

Theorem 13 (Reitz [8,9]). *The Ground Axiom is first order expressible in set theory.*

The Ground Axiom holds in many canonical models of set theory, such as L, $L[0^\sharp]$, $L[\mu]$ and many instances of K. Since these models all exhibit many highly regular structural features, it is very natural to inquire: To what extent are these regularity features consequences of the Ground Axiom? The answer, which Reitz provided in his dissertation, is that every model of ZFC has a class forcing extension, preserving any desired initial segment V_α (and mild in the sense that every new set is generic for set forcing), which is a model of the Ground Axiom. Thus, the Ground Axiom does not imply any of the usual combinatorial set-theoretic regularity features $\Diamond$, GCH and so on. Reitz's method obtained the Ground Axiom by forcing very strong versions of $V = \mathrm{HOD}$, and so his analysis did not settle the question of whether GA $\implies V = \mathrm{HOD}$. In a three-generation collaboration, we settled that question with the following:

Theorem 14 (Hamkins, Reitz, Woodin [5]). *Every model of set theory has an extension which is a model of* GA *plus* $V \neq \mathrm{HOD}$.

After some preparatory forcing, we use a class Silver iteration adding a Cohen subset to every regular cardinal. The argument is flexible and robust, and leads us to expect the Ground Axiom after most any Easton support progressively closed class iteration.

Let me set some terminology. A transitive class W is a *ground* of V if $W \models$ ZFC and $V = W[G]$ is a forcing extension of W by set forcing $G \subseteq \mathbb{P} \in W$. The model W is a *bedrock* of V if it is a ground of V and there is no deeper ground inside W. Equivalently, W is a bedrock of V if it is a ground of V and satisfies the Ground Axiom.

Theorem 15 (Reitz [8]). *If there is a model of* ZFC*, then there is a model of* ZFC *having no bedrock.*

We don't know if a model can have more than one bedrock model.

Question 16. *Is the bedrock unique when it exists?*

The principal new concept is the following:

Definition 17. The *Mantle* M of V is the intersection of all grounds of V.

The Mantle is a first-order parameter-free definable transitive class containing all ordinals. Much of this is easy to see, once one realizes that there is a broad uniformity in the definition of the ground model in its forcing extensions. The basic situation is described by the following.

Theorem 18. *There is a parameterized family* $\{ W_r \mid r \in V \}$ *of transitive classes such that*

1. *Every* W_r *is a ground of* V *and* $r \in W_r$.
2. *Every ground of* V *is* W_r *for some* r.
3. *The relation "*$x \in W_r$*" is first order.*
4. *The relation "*$V = W_r[G]$ *by* W_r*-generic filter* $G \subseteq \mathbb{P} \in W_r$*" is first order in the variables* $(r, G, \mathbb{P})$.

5. *The definition is somewhat absolute.*
   i. *If* $W_r \subseteq U \subseteq V$, *then* $W_r = W_r^U$.
   ii. *If* $V \subseteq V[G]$, *then* $\forall r \exists s\, W_r = W_s = W_s^{V[G]}$.

The parameterized family $\{\, W_r \mid r \in V \,\}$ of grounds in Theorem 18 reduces second order properties about grounds to first order properties about their parameters in this family. For example, the Ground Axiom holds if and only if $\forall r\, W_r = V$. The model W_r is a bedrock if and only if $\forall s\, (W_s \subseteq W_r \implies W_s = W_r)$. The Mantle is defined by $\mathrm{M} = \{\, x \mid \forall r\, (x \in W_r) \,\}$. Because of Theorem 18, each of these assertions is first order expressible in the language of set theory. The proof of Theorem 18 relies, of course, on the proof of Theorem 11, and I would like to mention a few of the ideas. Laver's proof of Theorem 11 relied on the following definitions and lemmas.

Definition 19 (Hamkins [2])

1. $W \subseteq V$ has the δ *covering property* if every $A \subseteq W$ with $A \in V$ and $|A|^V < \delta$ is covered $A \subseteq B$ by some $B \in W$ with $|B|^W < \delta$.
2. $W \subseteq V$ has the δ *approximation property* if every $A \subseteq W$ with $A \in V$ and all small approximations $A \cap B$ in W, whenever $|B|^W < \delta$, is already in the ground model $A \in W$.

Lemma 20 (Hamkins [2]). *If* $V \subseteq V[G]$ *and* $G \subseteq \mathbb{P} * \dot{\mathbb{Q}}$ *is* V*-generic for forcing with* $\mathbb{P}$ *nontrivial and* $\Vdash \dot{\mathbb{Q}}$ *is* $\leq |\mathbb{P}|$*-strategically closed, then* $V[G]$ *has the* δ *cover and approximation properties for* $\delta = |\mathbb{P}|^+$.

Lemma 21 (Laver [7], Hamkins). *If* $W, W' \subseteq V$ *have the* δ *approximation and covering properties,* $P(\delta)^W = P(\delta)^{W'}$ *and* $(\delta^+)^W = (\delta^+)^{W'} = (\delta^+)^V$, *then* $W = W'$.

Laver had first proved Lemma 21 for small forcing, that is, replacing the δ approximation and covering properties with the assumption that the forcing had size less then δ (which by Lemma 20 is a special case), and I extended it to the approximation and covering properties. Lemma 21 essentially provides the definition of W inside the forcing extension $W[G]$, using the parameter $P(\delta)^W$.

When looking downward at the various grounds, it is very natural to inquire whether one can fruitfully intersect them. Let us define that the grounds are *downward directed* if for every r and s there is t such that $W_t \subseteq W_r \cap W_s$. The grounds are *locally downward directed* if for every B and every r, s there is t with $W_t \cap B \subseteq W_r \cap W_s$. The question of whether there can be distinct bedrock models in the universe is of course related to the question of whether there is a ground in their intersection:

Question 22. *Are the grounds downward directed?*

Generalizing beyond finite intersections, let us define that the grounds are *downward set-directed* if for every A there is t with $W_t \subseteq \bigcap_{r \in A} W_r$. The grounds are *locally downward set-directed* if for every A, B there is t with $W_t \cap B \subseteq \bigcap_{r \in A} W_r$.

Question 23. *Are the grounds downward set directed?*

In every model for which we can determine the answer to this question, the answer is yes. The importance of the question is that in the situations where the answwer is yes, the Mantle is well behaved.

Theorem 24

1. *If the grounds are downward directed, then the Mantle is constant across the grounds, and* $\mathrm{M} \models \mathrm{ZF}$.
2. *If the grounds are downward set-directed, then* $\mathrm{M} \models \mathrm{ZFC}$.

The hypothesis in (2) can be weakened to require only that the grounds are downward directed and locally downward set-directed. The general fact underlying Theorem 24 is the following, where we define that a family $\mathcal{W}$ of transitive models of ZFC is *locally realized* if for every $y \in \cap\mathcal{W}$ there is $W \in \mathcal{W}$ with $P(y)^{\cap\mathcal{W}} = P(y)^{W}$. That is, for any y in all the models, there is a particular model $W \in \mathcal{W}$ that computes $P(y)$ the same as $\cap\mathcal{W}$ does. This is actually equivalent to requiring for every ordinal α that there is some $W \in \mathcal{W}$ such that $V_\alpha^W = V_\alpha^{\cap\mathcal{W}}$.

Theorem 25. *If* $\mathcal{W}$ *is a collection of transitive models of* ZFC*, all with same ordinals, and* $\cap\mathcal{W}$ *is a class in each* $W \in \mathcal{W}$*, then:*

1. $\cap\mathcal{W} \models \mathrm{ZF}$.
2. *If* $\mathcal{W}$ *is locally realized, then* $\cap\mathcal{W} \models \mathrm{ZFC}$.

Are the grounds locally realized? It is not difficult to see that the grounds are locally realized if and only if they are locally downward set-directed. We are somewhat embarrassed not to know the answer to the following question.

Question 26. *Does the Mantle always satisfy* ZF*?* ZFC*?*

It is natural, of course, to consider how the Mantle is affected by forcing. Since every ground model of V is a ground model of any forcing extension $V[G]$, it follows that the Mantle of $V[G]$ is contained within the Mantle of V. That is, the Mantle gets smaller (or at least no larger) as you perform more and more forcing. In the limit of this process, we arrive at:

Definition 27. The *generic mantle* of a model of set theory V is the intersection of all ground models of all set forcing extensions of V.

We will use the notation M to denote the Mantle and gM to denote the generic Mantle. The generic Mantle has proved in several ways to be a more robust version of the Mantle (although in truth we do not know them to differ). For example, for any model of ZFC, the generic Mantle is always a model at least of ZF, without any need for a directedness hypothesis. If the *generic grounds*, the ground models of the forcing extensions of V, are downward directed, then in fact the Mantle and the generic Mantle are the same. If the generic grounds are downward set directed, then the generic Mantle is a model of ZFC.

Question 28. *Can the mantle and the limit mantle differ?*

One might hope to prove even that every model V of ZFC is the generic mantle of a model W of the Ground Axiom, so that the mantle of W is W. This would provide a very attractive answer to Question 28.

Set-theoretic geology is naturally carried out in a context that includes all the forcing extensions of a model of set theory, all the grounds of these extensions, all forcing extensions of these resulting grounds and so on. The *generic multiverse* of a model of set theory, introduced by Woodin [11], is the smallest family of models of set theory containing that model and closed under both forcing extensions and grounds. There are numerous philosophical motivations to study to the generic multiverse. Indeed, Woodin introduced it specifically in order to criticize a certain multiverse view of truth, namely, truth as true in every model of the generic multiverse. Although I do not hold such a view of truth, nevertheless I want to investigate the fundamental features of the generic multiverse, a task I place at the foundation of any deep understanding of forcing. Surely the generic multiverse is the most natural and illuminating background context for the project of set-theoretic geology.

The generic Mantle gM is a parameter-free uniformly definable class, invariant by forcing, containing all ordinals and gM $\models$ ZF. Because it is invariant by forcing, it follows that the generic Mantle gM is constant across the multiverse, and in fact, it follows that the generic Mantle gM is the intersection of the generic multiverse. On this view, the generic Mantle is a canonical, fundamental feature of the generic multiverse, deserving of intense study.

The class HOD is the class of hereditarily ordinal definable sets, the sets that are definable using ordinal parameters, and all their members are definable using ordinal parameters, and so on. Introduced classically, HOD is intensely studied, and known to be a transitive inner model of ZFC, containing all ordinals. Let me now define the *generic* HOD to be the intersection of all HODs of all the forcing extensions.

$$\mathrm{gHOD} = \bigcap_G \mathrm{HOD}^{V[G]}$$

The generic HOD was originally introduced by Fuchs in an attempt to identify a very large canonical forcing-invariant class.

Theorem 29

1. gHOD *is constant across the generic multiverse.*
2. *The* HOD*s of all forcing extensions are downward set-directed.*
3. *Consequently,* gHOD *is locally realized and* gHOD $\models$ ZFC*.*
4. *The following inclusions hold.*

$$\begin{array}{ccccc} \mathrm{HOD} & & & & \\ \cup & & & & \\ \mathrm{gHOD} & \subseteq & \mathrm{gM} & \subseteq & M \end{array}$$

Question 30. *To what extent can we control and separate these classes?*

For the remainder of this article, I will sketch several answers to this question. First, we can control the classes to keep them all low.

Theorem 31 (Fuchs, Hamkins, Reitz). *If* $V \models \mathrm{ZFC}$*, then there is a class extension* $V[G]$ *in which*

$$V = \mathrm{M}^{V[G]} = \mathrm{gM}^{V[G]} = \mathrm{gHOD}^{V[G]} = \mathrm{HOD}^{V[G]}$$

In particular, as mentioned earlier, every model of ZFC is the mantle and generic mantle of another model of ZFC. It follows that we cannot expect to prove any regularity features about the mantle or the generic mantle, beyond what can be proved about an arbitrary model of ZFC. It also follows that the mantle of V is not necessarily a ground model of V, even when it is a model of ZFC. One can therefore iteratively take the mantle of the mantle and so on, and we have proved that this process can strictly continue. Indeed, by iteratively computing the Mantle of the Mantle and so on, what we call the *inner* Mantles, we might eventually arrive at a model of ZFC plus the Ground Axiom, where the process would naturally terminate. If this should occur, then we call this termination model the *outer core* of the original model. Generalizing the theorem above, Fuchs, Reitz and I have conjectured that every model of ZFC is the α^{th} inner Mantle of another model of ZFC, for arbitrary ordinals α (or even $\alpha = \mathrm{ORD}$ or beyond).

There is an interesting philosophical view related to and perhaps refuted by this conjecture, namely, the philosophical view holding that there is a highly regular core underlying the universe of set theory, an inner model obscured over the eons by the accumulating layers of debris heaped up by innumerable forcing constructions since the beginning of time. If only we could sweep this accumulated material away, the view holds, then we should find an ancient paradise. The Mantle, of course, wipes away an entire strata of forcing, and the iterated inner Mantles sweep away additional layers. So the philosophical view would lead us to believe that in this way we would be getting close to a highly regular core. If the conjecture is correct, however, then what we should expect to find after sweeping such layers away even ORD many times is, not a highly regular ancient paradise, but rather something ordinary: an arbitrary model of set theory.

Let me sketch a few ideas from the proof of Theorem 31. The initial idea goes back to McAloon (1971), who explained how to make sets definable by forcing. For an easy warm-up case, consider an arbitrary real $x \subseteq \omega$. This real may not happen to be definable in V, but it is an elementary exercise in product forcing to force the GCH to hold at $\aleph_n$ exactly when $x(n) = 1$. In the resulting forcing extension $V[G]$, therefore, the original real x is definable, without parameters. In a similar way, any set whatsoever can become definable in a forcing extension. For the main theorem, we start in $V \models \mathrm{ZFC}$, and seek a class forcing extension $V[G]$ with $V = \mathrm{M}^{V[G]} = \mathrm{gM}^{V[G]} = \mathrm{gHOD}^{V[G]} = \mathrm{HOD}^{V[G]}$. Let $\mathbb{Q}_\alpha$ generically decide whether to force GCH or $\neg$GCH at $\aleph_\alpha$ (the actual proof is somewhat more complicated than this), and let $\mathbb{P} = \Pi_\alpha \mathbb{Q}_\alpha$ be the class sized product, with

set support. The desired model will be the class forcing extension $V[G]$, where $G \subseteq \mathbb{P}$ is V-generic. First, every set in V becomes coded unboundedly into the continuum function of $V[G]$, because it is dense that the generic filter will opt to code it into the GCH pattern. Therefore, every set in V becomes definable in $V[G]$ and all set forcing extensions of $V[G]$. This establishes $V \subseteq \mathrm{gHOD}$ and consequently $V \subseteq \mathrm{gHOD} \subseteq \mathrm{gM} \subseteq \mathrm{M}$ and $V \subseteq \mathrm{HOD}$. Next, observe that every tail segment $V[G^\alpha]$, using only the part of the forcing beyond α, is a ground of $V[G]$. By mutual genericity of the forcing up to α with the forcing at stages after α, it follows that $\cap_\alpha V[G^\alpha] = V$. This implies that $\mathrm{M} \subseteq V$ and consequently $V = \mathrm{gHOD} = \mathrm{gM} = \mathrm{M}$. Finally, $\mathrm{HOD}^{V[G]} \subseteq \mathrm{HOD}^{V[G^\alpha]}$, since $\mathbb{P} \restriction \alpha$ is densely almost homogeneous. It follows that $\mathrm{HOD}^{V[G]} \subseteq V$. In summary, $V = \mathrm{M}^{V[G]} = \mathrm{gM}^{V[G]} = \mathrm{gHOD}^{V[G]} = \mathrm{HOD}^{V[G]}$, as desired.

Let me now turn to a second answer to Question 30, where we keep the Mantles low, while allowing HOD to inflate.

Theorem 32 (Fuchs, Hamkins, Reitz). *If $V \models \mathrm{ZFC}$, then there is a class extension $V[G]$ in which*

$$V = \mathrm{M}^{V[G]} = \mathrm{gM}^{V[G]} = \mathrm{gHOD}^{V[G]} \qquad \textit{but} \quad \mathrm{HOD}^{V[G]} = V[G]$$

For this theorem, our strategy is to balance the forces on M, gM, gHOD and HOD. We perform proper class forcing to $V[G]$ in such a way that every set in V will be coded unboundedly into the GCH pattern, and we also ensure that G itself is definable, but not so robustly, so that the gHOD will fall back down to V. Specifically, the proof uses many instances of *self-encoding forcing*, the set-sized forcing which first adds a Cohen subset $A \subseteq \kappa$, and then codes this new set A into the GCH pattern above κ, and then codes the resulting new sets into the next block of the GCH pattern, and then those sets, and so on. By the next Beth fixed point above κ, we find an extension $V[G_{(\kappa)}]$ in which $G_{(\kappa)}$ is definable. To prove the theorem, one takes a class-sized product of such self-encoding forcing, which operate on non-interfering intervals of cardinals. The result is a class forcing extension $V[G]$ in which the Mantle and the generic Mantle and the generic HOD are V, but the HOD is $V[G]$. The reason the generic HOD falls back down to V is that with subsequent collapse forcing, one can in effect erase the coding of any given $G_{(\alpha)}$, and so the generic HOD of $V[G]$, and indeed the generic Mantle and Mantle, is once again contained in the intersection of the tail forcing extensions.

Next, we keep the HODs low, while allowing the Mantle to inflate, seeking $V[G]$ with $V = \mathrm{HOD}^{V[G]} = \mathrm{gHOD}^{V[G]}$ but $\mathrm{M}^{V[G]} = V[G]$. Such a model $V[G]$ will of course be a model of the Ground Axiom plus $V \neq \mathrm{HOD}$. Recall Theorem 3, which says that every $V \models \mathrm{ZFC}$ has a class forcing extension $V[G] \models \mathrm{GA} + V \neq \mathrm{HOD}$. By modifying the argument, we are able to obtain:

Theorem 33 (Fuchs, Hamkins, Reitz). *If $V \models \mathrm{ZFC}$, then there is a class extension $V[G]$ in which*

$$V = \mathrm{HOD}^{V[G]} = \mathrm{gHOD}^{V[G]} \qquad \textit{but} \quad \mathrm{M}^{V[G]} = V[G]$$

We have not yet been able to compute the generic Mantle of this model. Our last combination is to push both the Mantles and the HODs high.

Theorem 34 (Fuchs, Hamkins, Reitz). *If $V \models$ ZFC, then there is a class forcing extension $V[G]$ in which*

$$V[G] = \mathrm{HOD}^{V[G]} = \mathrm{gHOD}^{V[G]} = \mathrm{M}^{V[G]} = \mathrm{gM}^{V[G]}$$

This theorem is proved by Reitz's method of forcing every set to be coded into the GCH pattern. I would like to emphasize that in none of our theorems have we managed to separate the generic Mantle from either the Mantle or the generic HOD. We know that gHOD $\subseteq$ gM $\subseteq$ M, and we have separated the generic HOD from the Mantle in Theorem 32, so the model of this theorem does perform at least one of the desired separations, but as we have not been able to compute the generic Mantle of that model, we don't know which separation has occurred. Thus, I reiterate Question 28 in part, in the dual formulation.

Question 35. *Is the generic Mantle the same as the Mantle? Is the generic Mantle the same as the generic* HOD*?*

Of course, not both answers can be yes, and we expect that both answers are no. Let me close the article by inviting all researchers to attack this open question and the others I have mentioned. The research topics here are young and ripe for progress.

References

1. Fuchs, G., Hamkins, J.D., Reitz, J.: Set-theoretic geology (in preparation)
2. Hamkins, J.D.: Extensions with the approximation and cover properties have no new large cardinals. Fundamenta Mathematicae 180(3), 257–277 (2003)
3. Hamkins, J.D.: A simple maximality principle. Journal of Symbolic Logic 68(2), 527–550 (2003)
4. Hamkins, J.D., Löwe, B.: The modal logic of forcing. Transactions of the American Mathematical Society 360, 1793–1817 (2008)
5. Hamkins, J.D., Reitz, J., Woodin, W.H.: The Ground Axiom is consistent with V $\neq$ HOD. Proceedings of the AMS 136, 2943–2949 (2008)
6. Hamkins, J.D., Seabold, D.: Boolean ultrapowers (in preparation)
7. Laver, R.: Certain very large cardinals are not created in small forcing extensions. Annals of Pure and Applied Logic 149(1–3), 1–6 (2007)
8. Reitz, J.: The Ground Axiom. Ph.D thesis, The Graduate Center of the City University of New York (September 2006)
9. Reitz, J.: The Ground Axiom. Journal of Symbolic Logic 72(4), 1299–1317 (2007)
10. Stavi, J., Väänänen, J.: Reflection principles for the continuum. In: Zhang, Y. (ed.) Logic and Algebra. AMS Contemporary Mathematics Series, vol. 302 (2003)
11. Hugh Woodin, W.: Recent development's on Cantor's Continuum Hypothesis. In: Proceedings of the Continuum in Philosophy and Mathematics. Carlsberg Academy, Copenhagen (November 2004)

Connection Matrices for MSOL-Definable Structural Invariants

Johann Makowsky[⋆]

Department of Computer Science
Technion–Israel Institute of Technology, Haifa, Israel
janos@cs.technion.ac.il

Abstract. Connection matrices of graph parameters were first introduced by M. Freedman, L. Lovász and A. Schrijver (2007) to study the question which graph parameters can be represented as counting functions of weighted homomorphisms. The rows and columns of a connection matrix $M(f, \Box)$ of a graph parameter f and a binary operation $\Box$ are indexed by all finite (labeled) graphs G_i and the entry at (G_i, G_j) is given by the value of $f(G_i \Box G_j)$. Connection matrices turned out to be a very powerful tool for studying graph parameters in general.

B. Godlin, T. Kotek and J.A. Makowsky (2008) noticed that connection matrices can be defined for general relational structures and binary operations between them, and for general structural parameters. They proved that for structural parameters f definable in Monadic Second Order Logic, (*MSOL*) and binary operations compatible with *MSOL*, the connection matrix $M(f, \Box)$ has always finite rank. In this talk we discuss several applications of this Finite Rank Theorem, and outline ideas for further research.

1 Introduction

Graph Parameters and Graph Polynomials. A graph parameter (also called a numeric graph invariant) f is a function from the class of all finite graphs $\mathcal{G}$ to some numeric domain which is an ordered commutative ring $\mathcal{R}$ or an ordered field $\mathcal{F}$ with 0 and 1, usually the integers $\mathbb{Z}$, the rational numbers $\mathbb{Q}$ or the reals $\mathbb{R}$. Graph properties are the special case where the values of f are 0 or 1. In the case of graph properties the ring can be taken to be the two-element boolean algebra, or, alternatively the field $\mathbb{Z}_2$. We shall use the latter, to make our use of linear algebra uniform.

Graph polynomials are functions p from $\mathcal{G}$ into a polynomial ring, usually $\mathbb{Z}[\bar{X}]$, where $\bar{X}$ is a fixed finite set of indeterminates. Graph polynomials are a way to encode infinitely many graph parameters. Every evaluation of the polynomial $p(G; \bar{X})$ at some point $\bar{X} = \bar{x_0}$ is a graph parameter. So are the coefficients

[⋆] Partially supported by the Israel Science Foundation for the project "Model Theoretic Interpretations of Counting Functions" (2007-2010) and the Grant for Promotion of Research by the Technion–Israel Institute of Technology.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 51–64, 2009.

of $p(G;\bar{X})$, the total degree or the degree of monomials where the coefficient satisfies certain properties, and the zeros of $p(G;\bar{X})$.

Instead of graphs one can also consider hypergraphs or relational structures over some fixed finite vocabulary τ, a set consisting of relation symbols and constants. In this case we speak of *structural invariants for τ-structures*, or just of *τ-invariants* and *τ-polynomials*. We include here the empty τ-structure, which we denote by $\emptyset_\tau$.

Connection Matrices. Let $\Box$ be a binary operation on τ-structures (which respects τ-isomorphisms. A τ-structure $\mathcal{I}$ is $\Box$-neutral if for every τ-structure $\mathcal{A}$ we have $\mathcal{A}\Box\mathcal{I} \simeq \mathcal{I}\Box\mathcal{A}$. For the disjoint union of τ-structures, denoted by $\sqcup$, the empty structure is $\sqcup$-neutral. For the cartesian product of τ-structures, denoted by $\times$, the one-element structure with full relations is $\times$-neutral.

Let f be a τ-invariant and $\Box$ be a binary operation on τ-structures which respects τ-isomorphisms. Let $\{\mathcal{A}_i : i \in \mathbb{N}\}$ be an enumeration of all finite τ-structures (up to isomorphisms). We define the infinite matrix

$$M(f,\Box) = (m_{i,j}(f,\Box))$$

by $m_{i,j}(f,\Box) = f(\mathcal{A}_i\Box\mathcal{A}_j)$. $M(f,\Box)$ is called the *connection matrix of f and $\Box$*. We denote by $r_{\mathcal{R}}(f,\Box)$ the rank over $\mathcal{R}$ of the matrix $M(f,\Box)$. We usually omit the subscript in $r_{\mathcal{R}}(f,\Box)$, when no confusion arises.

Multiplicative $\boldsymbol{\tau}$-invariants. A τ-invariant f is called $\Box$-*multiplicative* if it satisfies $f(\mathcal{A}\Box\mathcal{B}) = f(\mathcal{A})\cdot f(\mathcal{B})$ for all finite τ-structures $\mathcal{A}$ and $\mathcal{B}$. With respect to the disjoint union, $\sqcup$, typical examples of $\sqcup$-multiplicative graph parameters are $\chi(G,k)$, the number of proper vertex colorings with k colors, $pm(G)$, the number of perfect matchings, or the number of acyclic orientations. With respect to the join of graphs, denoted by $\bowtie$, counting the number of covers by independent sets is $\bowtie$-multiplicative.

In [FLS07] the following characterization of graph parameters, which are multiplicative with respect to the disjoint union, was given. We state it here in the general context. The proof is verbatim the same.

Proposition 1. *Let f be a τ-invariant with values in an ordered field $\mathcal{R}$, and which is not identically 0, and let $\Box$ be a binary operation on τ-structures, with $\mathcal{I}$ being a unique $\Box$-neutral structure. Then f is $\Box$-multiplicative if and only if $f(\mathcal{I}) = 1$, and the matrix $M(f,\Box)$ has rank 1 and is positive semi-definite.*

Connection matrices for various operations on labeled graphs are studied in [FLS07, Sze07, Lov07, Sch08]. In these papers they are used to characterize graph invariants arising from various vertex-coloring and edge-coloring models. In [GKM08] connection matrices are used to study definability properties of graph parameters and graph polynomials.

Outline of the Talk. In this talk we summarize the results from [FLS07] and [GKM08] and discuss further applications and open problems. In Section 2 we introduce connection matrices of τ-invariants and their rank. We illustrate

their uses in the case of graph parameters. This paraphrases the main results of [FLS07], and explains the Freedman-Lovász-Schrijver Theorem which gives a characterization of graph parameters arising from counting weighted homomorphisms. In Section 3 we show how one can use a Feferman-Vaught-style Theorem from [Mak04] for τ-invariants definable in Monadic Second Order Logic *MSOL* to show that the rank of many connection matrices has to be finite. The exact statement of this is the Finite Rank Theorem (Theorem 9). In Section 4 we give applications of the Finite Rank Theorem, mostly taken from [GKM08]. We conclude with a list of open problems for further investigations.

2 Properties of Connection Matrices of τ-Invariants

The Rank of Connection Matrices. Besides multiplicative τ-invariants we consider also τ-invariants f with the following properties:

(i) f is $\Box$-*additive* if $f(G_1 \Box G_2) = f(G_1) + f(G_2)$.

(ii) f is $\Box$-*maximizing*, respectively $\Box$-*minimizing* if there exist infinite sequences of graphs $(G_i)_{i \in \mathbb{N}}, (H_i)_{i \in \mathbb{N}}$ such that for all $(i,j) \in \mathbb{N}^2$ we have

$$f(G_i \Box H_j) = \max\{f(G_i), f(H_j)\},$$

respectively

$$f(G_i \Box H_j) = \min\{f(G_i), f(H_j)\}.$$

Furthermore, for all $i \in \mathbb{N}$ the sequence $f(i)_j = f(G_i \Box H_j)$ is strictly monotone increasing. If the two sequences consist of all τ-structures, we say f is *strictly* $\Box$-*maximizing*, respectively *strictly* $\Box$-*minimizing*.

(iii) A τ-invariant f is *weakly* $(\Box, \gamma)$-*multiplicative*, if there exists a finite set of graph parameters $f_i : i \leq \gamma$ with $i, \gamma \in \mathbb{N}$ with $f = f_0$, and a matrix $N \in \mathbb{R}^{\gamma \times \gamma}$, such that $f_0(\mathcal{A}_1 \Box \mathcal{A}_2) = \sum_{i,j} f_i(\mathcal{A}_1) N_{ij} f_j(\mathcal{A}_2)$.

In other words, $f(\mathcal{A}_1 \Box \mathcal{A}_2)$ is given by a quadratic form defined by $N_{i,j}$ of rank at most γ.

Typical examples[1] of $\sqcup$-additive parameters are the cardinality of the vertex set, the cardinality of the edge-set, $k(G)$, $b(G)$, number of connected components and number of blocks (doubly connected components), respectively. Examples of $\bowtie$-additive graph parameter are $\chi(G)$ and $\omega(G)$. Among the $\sqcup$-maximizing graph parameters we have: the chromatic number $\chi(G)$, the edge chromatic number $\chi_e(G)$, and the total coloring number $\chi_t(G)$, the size of a maximal clique $\omega(G)$, the maximal degree $\Delta(G)$, the tree-width $tw(G)$, and the clique-width $cw(G)$.

Proposition 2. *Let f be a τ-invariant.*

(i) If f is $\Box$-multiplicative, $r(f, \Box) = 1$.
(ii) If f is $\Box$-additive, $r(f, \Box) = 2$, unless $M(f, \Box)$ is the zero matrix.

[1] Almost all graph parameters discussed are taken from [Die96]. One exception is the clique-width, which was introduced in [CO00], and, in connection to graph polynomials, in [Mak04].

(iii) *If f is $\Box$-maximizing or $\Box$-minimizing, $r(f, \Box)$ is infinite.*
(iv) *Let f be a graph parameter which is weakly $(\Box, \gamma)$-multiplicative. Then $r(f, \Box) \leq \gamma$.*

Proof. (i) was already stated in Proposition 1. (ii), (iii) and (iv) are easy to verify.

Counting Weighted Homomorphisms of Graphs. A k-graph is a graph $G = (V(G), E(G))$ with k distinct vertices labeled with $0, 1, \ldots, k-1$. We denote by $\mathcal{G}_k$ the class of finite k-graphs. $\mathcal{G}_0 = \mathcal{G}$ the set of all finite graphs without labels.

Given two k-graphs G_1, G_2 we define the k-sum $G_1 \sqcup_k G_2$ as the disjoint union of G_1 and G_2 where we identify correspondingly labeled vertices. In [FLS07] the connection matrices $M(f, \sqcup_k)$ on $\mathcal{G}_k$ are used to characterize those graph parameters f which can be represented as counting functions of weighted homomorphisms. The setup is as follows:

Let $H = (V(H), E(H)) \in \mathcal{G}$ be a fixed graph, possibly with loops. Let $\alpha : V(H) \to \mathbb{R}^+$ and $\beta : E(H) \to \mathbb{R}$ be weight functions of vertices and edges respectively, and let $h : G \to H$ be a homomorphism. We define weights of h by

$$\alpha_h = \prod_{u \in V(G)} \alpha(h(u)) \quad \text{and} \quad \beta_h = \prod_{u,v \in E(G)} \beta(h(u), h(v))$$

Finally, we sum over all homomorphisms

$$Z_{H,\alpha,\beta}(G) = \sum_{h:G \to H} \alpha_h \cdot \beta_h.$$

$Z_{H,\alpha,\beta}(G)$ is often called a *partition function* or a *vertex coloring model.*

Observation 1. *Partition functions are $\sqcup$-multiplicative.*

Example 1. *The following are simple partition functions:*

(i) *For $H = K_m$, a clique with m vertices,*

$$Z_{K_m,1,1}(G) = \chi(G, m)$$

which counts the number of proper m-colorings.

(ii) *For $H = L_1$, an isolated loop, $\alpha = \lambda$, $\beta = \mu$,*

$$Z_{L_1,\lambda,\mu}(G) = \lambda^{|V(G)|} \cdot \mu^{|E(G)|}$$

(iii) *For $H = L_m$ consisting of m isolated loops, $\alpha = \lambda$, $\beta = \mu$,*

$$Z_{L_m,\lambda,\mu}(G) = m^{k(G)} \cdot \lambda^{|V(G)|} \cdot \mu^{|E(G)|}$$

(iv) *For $H = K_1 \bowtie L_1$ with vertices v, ℓ respectively, and $\alpha(v) = X, \alpha(\ell) = 1$, $\beta = 1$ we get*

$$Z_{K_1 \bowtie L_1,\alpha,\beta}(G) = \sum_i ind_i(G) \cdot X^i$$

where $ind_i(G)$ is the number of independents sets of size i in G.

In [FLS07] it is proved that the connection matrices $M(f, \sqcup_k)$ for $f = Z_{H,\alpha,\beta}(G)$ have the following properties:

Proposition 3 (M. Freedman, L. Lovász and A. Schrijver, 2007)

(i) For every weighted graph (H, α, β) we have

$$r(Z_{H,\alpha,\beta}(G), \sqcup_k) \leq |V(H)|^k$$

(ii) If (H, α, β) has no automorphisms and no twins, then

$$r(Z_{H,\alpha,\beta}(G), \sqcup_k) = |V(H)|^k$$

Automorphisms here are weight preserving. Two vertices $u, v \in V(H)$ of (H, α, β) are *twins* if for every $w \in V(H)$ we have that $\beta(u, w) = \beta(v, w)$. Being twins does not depend on α.

Proposition 4 (M. Freedman, L. Lovász and A. Schrijver, 2007)

For every weighted graph (H, α, β) the matrix $M(Z_{H,\alpha,\beta}(G), \sqcup_k)$ is positive semi-definite.

Example 2. *(i) Let $pm(G)$ denote the number of perfect matchings of G. $pm(G)$ is $\sqcup$-multiplicative and $r(pm, \sqcup_k) = 2^k$, but $M(pm, \sqcup_1)$ is not positive definite.*
(ii) For $\chi(-, \lambda)$, $\lambda \in \mathbb{Z}$ we have: $M(\chi(-, \lambda), \sqcup_k)$ is positive-semi-definite, and $r(\chi(-, \lambda), \sqcup_k)$ is finite, but exponentially bounded only for $\lambda \in \mathbb{Z}^+$, otherwise it grows superexponentially.

The Freedman-Lovász-Schreijver Theorem. We say that a numeric graph invariant is *hom-presentable* if there is a weighted graph (H, α, β) such that for every G $f(G) = Z_{H,\alpha,\beta}(G)$. We have seen in Example 1 that $2^{|V(G)|}, 2^{|E(G)|}, 2^{k(G)}$ are hom-presentable, but by Proposition 2 and 3, $|V(G)|, |E(G)|, k(G)$ are not hom-presentable, as their connection matrices have infinite rank. $\chi(-, \lambda)$ is hom-representable for every $\lambda \in \mathbb{Z}^+$, but the *choice of (H, α, β) depends on* λ.

Theorem 5 (M. Freedman, L. Lovász and A. Schrijver, 2007)
Let f be a real-valued graph parameter. f is hom-presentable iff for every $k \in \mathbb{N}$

(i) $M(f, \sqcup_k)$ is positive semi-definite, and
(ii) $r(f, \sqcup_k) \leq q^k$ for some $q \in \mathbb{N}^+$.

There are various generalizations of Theorem 5. B. Szegedy [Sze07] considers *edge coloring models* and connection matrices $S(f, k)$ based on identification of k unfinished edges. A. Schrijver [Sch08] unifies the proofs of [FLS07] and [Sze07] using further variations of connection matrices defined also for hyper-graphs and directed graphs.

3 Enter Logic

Monadic Second Order Logic. A vocabulary is a finite set of relation and constant symbols. We define the logic *MSOL* for τ-structures inductively. We have first order variables $x_i : i \in \mathbb{N}$ which range over elements of A, the universe of a τ-structure, and (monadic) second order variables $U_i : i \in \mathbb{N}$, which range over subsets of A. Terms $t, t', \ldots$ are either first order variables or constant symbols from τ. Atomic formulas are of the form $t = t'$, $R(\bar{t})$, where R is a relation symbol of τ $U_i(t)$ and have the natural interpretation. Formulas are built inductively using the connectives $\vee, \wedge, \rightarrow, \leftrightarrow, \neg$, and the quantifiers $\forall x_i$, $\exists x_i$, $\forall U_i$, $\exists U_i$ with their natural interpretation. The quantifier rank of an *MSOL*-formula ϕ is defined as usual and denoted by $qr(\phi)$ and for the rank we do not distinguish between first order and second variables.

***MSOL*-definable τ-Polynomials in Normal Form.** A *MSOL*-definable polynomial in indeterminates $X_1, \ldots, X_\ell$ in *normal form* has the form

$$\sum_{U_1:\Phi_1(U_1)} \sum_{U_2:\Phi_2(U_2)} \cdots \sum_{U_{\ell_1}:\Phi_{\ell_1}(U_{\ell_1})} \left(\prod_{\bar{x}_1:\phi_1(\bar{x}_1)} X_1 \prod_{\bar{x}_2:\phi_2(\bar{x}_2)} X_2 \ldots \prod_{\bar{x}_\ell:\phi_\ell(\bar{x}_\ell)} X_\ell \right)$$

where all the formulas Φ_i and ϕ_i are *MSOL*-formulas with the iteration variables (for summation and products) indicated. There may be additional parameters in the formulas. However, Φ_i may not contain the variables U_j for $j > i$, and ϕ_i may not contain $\bar{x}_j$ for $j > i$. Both Φ_i and ϕ_i are referred to as iteration formulas.

Looking at the partition function

$$Z_{H,\alpha,\beta}(G) = \sum_{h:G\rightarrow H} \alpha_h \cdot \beta_h. \tag{1}$$

we can rewrite it as follows: Let $G = (V(G), E(G))$, $H = (V(H), G(H))$ and $V(H) = \{v_0, \ldots, v_{n-1}\}$. We introduce, for each $v_i : i \leq n-1$ a set variable U_i. Let $\phi_{hom(H)}(U_0, \ldots, U_{n-1})$ be the formula $U_0, \ldots, U_{n-1}$ is a partition of $V(G)$ and that for all $x, y \in V(G)$, if $(x, y) \in E(G)$ then there is a $(v_i, v_j) \in E(H)$ such that $x \in U_i$ and $y \in U_j$. The formula $\phi_{hom(H)}(U_0, \ldots, U_{n-1})$ is a first order formula over the relation symbols for $E(G)$ and $U_0, \ldots, U_{n-1}$. It can also be viewed as a formula in Monadic Second Order Logic *MSOL* over the vocabulary consisting only of the binary relation symbol for $E(G)$.

Now the expression (1) can be, using $\bar{U} = (U_0, U_1, \ldots U_{n-1})$, written as

$$Z_{H,\alpha,\beta}(G) = \sum_{\bar{U}:\phi_{hom(H)}} \left(\left(\prod_{i=0}^{n-1} \prod_{x\in U_i} \alpha(x)) \right) \left(\prod_{(j,k)\in E(H)} \prod_{(y\in U_j \wedge z\in U_k)} \beta(y,z)) \right) \right) \tag{2}$$

If we consider all the $\alpha(v_i)$ and $\beta(v_j, v_k)$ as indeterminates, the left hand side of the expression (2) is a typical instance of a *MSOL*-definable graph polynomial

introduced in [Mak04]. For fixed values of $\alpha(v_i)$ and $\beta(v_j, v_k)$ this gives an *MSOL*-definable graph parameter, and, more generally, if we replace graphs by relational structures, of *MSOL*-definable τ-invariants. Hence we have shown:

Proposition 6. *For every α, β the graph parameter $Z_{H,\alpha,\beta}(G)$ is an MSOL-definable τ_1-invariant with $\tau_1 = \{E\}$.*

Using Finite Rank to Compute Partition Functions. Let $TW(k)$ and $CW(k)$ denote the class of graphs of tree-width and clique-width at most k, respectively. It was shown in [CO00] that $TW(k) \subseteq CW(2^{k+1}+1)$. Using the main results of [CMR01, Mak04] combined with [Oum05] we get from Proposition 6 the following complexity result.

Proposition 7. *On the class $CW(k)$ of graphs of clique-width at most k the graph invariants $Z_{H,\alpha,\beta}(G)$ can be computed in polynomial time, and are fixed parameter tractable, i.e., the exponent of the polynomial is independent of k, but the estimates obtained for the upper bounds for the constants are simply exponential in the case of $TW(k)$, but doubly exponential in k in the case of $CW(k)$.*

For graphs in $TW(k)$ this was already observed in [Lov07]. To get the better bound on the constants in the case of $TW(k)$, we can use Proposition 3 in the dynamic programming algorithm underlying the proofs in [CMR01, Mak04].

***MSOL*-compatible Operations on $\boldsymbol{\tau}$-structures.** Two τ-structures $\mathcal{A}, \mathcal{B}$, are said to be k-equivalent for *MSOL*, if they satisfy the same *MSOL*-sentences of quantifier rank k. We denote this equivalence relation by $\mathcal{A} \equiv_k \mathcal{B}$.

A binary operation $\Box$ on τ-structures is called *MSOL-k-compatible* if for $k \in \mathbb{N}$ we have that $\mathcal{A} \equiv_{m+k} \mathcal{A}'$ and $\mathcal{B} \equiv_{m+k} \mathcal{B}'$ implies that

$$\mathcal{A} \Box \mathcal{B} \equiv_m \mathcal{A}' \Box \mathcal{B}'.$$

The operation $\Box$ is called *MSOL-compatible* if there is some $k \in \mathbb{N}$ such that $\Box$ is *MSOL*-k-compatible.

In [Mak04] the case of $k = 0$ is called *MSOL-smooth*. The disjoint union of τ-structures is *MSOL*-smooth. So are the operations $\sqcup_k$ on k-graphs. The cartesian product $\times$ is not *MSOL*-compatible. However, the notion of *MSOL*-compatible operation is sensitive to the choice of the representation of, say, graphs as τ-structures. If we represent graphs $G = (V(G), E(G))$ as τ_1-structures with $\tau_1 = \{E\}$, which have universe $V(G)$ and a binary relation $E(G)$, the join operation $G_1 \bowtie G_2$ is *MSOL*-smooth. This is so, because it can be obtained from the disjoint union by the application of a quantifierfree transduction. If, however, we represent graphs as a two-sorted $\{R\}$ τ_2-structures, with $\tau_2 = \{P_V, P_E, R\}$, with sorts $P_V = V(G)$ and $P_E = E(G)$, and a binary incidence relation $R(G) \subset V(G) \times E(G)$, then $G_1 \bowtie G_2$ contains the cartesian product $V(G_1) \times V(G_2)$ in $E(G_1 \bowtie G_2)$ and behaves more like a cartesian product, which is not even *MSOL*-compatible. It is important to note that the operations $\sqcup_k$ are *MSOL*-smooth for graphs as τ_1-structures and as τ_2-structures.

The following theorem is proven in [Mak04, Theorem 6.4]:

Theorem 8. *Let f be a graph parameter which is the evaluation $f(G)=p(G,\bar{x}_0)$ of an MSOL-definable τ-polynomial $p(G,\bar{X})$. Furthermore, let $\Box$ be a binary operation on τ-structures which is MSOL-k-compatible. Then f is weakly $(\Box,\gamma))$-multiplicative for some $\gamma \in \mathbb{N}$ which depend on τ, the polynomial p, k, but not on $\bar{x}_0$.*

The Finite Rank Theorem. As in [GKM08], we get immediately, using Proposition 2 and Theorem 8 the following Theorem.

Theorem 9 (Finite Rank Theorem). *Let $p(G,\bar{X})$ be an MSOL-definable τ-polynomial with values in $\mathbb{R}[\bar{X}]$ with m indeterminates, and let $\Box$ be a binary operation on τ-structures which is MSOL-k-compatible. There is $\gamma_{\tau,\Box}(p) \in \mathbb{N}$ depending on τ, the polynomial p, and k only, such that for all $\bar{x}_0 \in \mathbb{R}^m$, we have $r(p(G,\bar{x}_0),\Box) \leq \gamma_{\tau,\Box}(p)$.*

The upper bound on the rank obtained in Theorem 9 again is very large. In the case of partition functions this bound is computed precisely in Proposition 3.

4 Applications of the Finite Rank Theorem

4.1 Non-definability in *MSOL*

Counting hamiltonian circuits. We shall look at the graph parameter $hc(G)$ which counts the number of hamiltonian circuits of a graph G, and the graph property HAM, which consists of all graphs which do have a hamiltonian circuit. If we represent graphs $G = (V(G), E(G))$ as τ_1-structures with $\tau_1 = \{E\}$, which have universe $V(G)$ and a binary relation $E(G)$, it is well known, cf. [dR84], that HAM is not *MSOL*-definable. If, however, we represent graphs as a two-sorted $\{R\}$ τ_2-structures, with $\tau_2 = \{P_V, P_E, R\}$, with sorts $P_V = V(G)$ and $P_E = E(G)$, and a binary incidence relation $R(G) \subset V(G) \times E(G)$, HAM is *MSOL*-definable.

Let E_m be the graph with m vertices and no edges. It is easy to see that $E_m \bowtie E_n$ contains exactly one hamiltonian circuit if and only if $m = n$. Therefore, $M(hc, \bowtie)$ and $M(HAM, \bowtie)$ both contain the infinite unit matrix as a submatrix, and $r(hc, \bowtie)$ is infinite over $\mathbb{Q}$, whereas $r(HAM, \bowtie)$ is infinite over $\mathbb{Z}_2$. We conclude that, HAM is not an *MSOL*-definable property of τ_1-structures, and that hc is not an evaluation of an τ_1-polynomial.

The subtle point is, that the join of two graphs is *MSOL*-smooth only for graphs as τ_1-structures. In the presentation as τ_2-structures, the sort $E(G_1 \bowtie G_2)$ grows quadratically in the size of $V(G_1)$ and $V(G_2)$, and is not even *MSOL*-compatible.

Graph colorings with no large monochromatic components. The same happens with the chromatic polynomial, and its relatives, the polynomials $mcc_t(G,k)$ for $t \in \mathbb{N} - \{0\}$. Following [LMST07], we denote by $mcc_t(G,k)$ the

number of functions $f : V(G) \to [k]$ such that for each $i \leq k$, the set $f^{-1}(i)$ induces a graph which consist of connected components of size at most t. Clearly, we have $\chi(G,k) = mcc_1(G,k)$. It follows from results in [KMZ08] that for each $t \in \mathbb{N}$ the counting function $mcc_t(G,k)$ is a polynomial in k.

Proposition 10 (T. Kotek). *For each $t \in \mathbb{N} - \{0\}$ the rank $r(mcc_t(G,k), \bowtie)$ tends to infinity with k.*

Corollary 11. *The polynomial $mcc_t(G,k)$ is not a τ_1-polynomial.*

But for connected graphs, we have $\chi(G,k) = T(G; 1-k, 0)$, where $T(G,X,Y)$ is the Tutte polynomial, which is *MSOL*-definable over the vocabulary $\tau_3 = \tau_2 \cup \{<_E\}$, where $<_E$ is a linear ordering of $E(G)$.

4.2 Evaluations of Well Known Graph Polynomials

A particular graph polynomial is considered interesting if it encodes many useful graph parameters. Let $G = (V(G), E(G))$ be a graph. The characteristic polynomial $P(G,X)$ of a graph is defined as the characteristic polynomial (in the sense of linear algebra) of the adjacency matrix A_G of G. The coefficients of $P(G,X)$ are defined by

$$det(X \cdot \mathbf{1} - A_G) = \sum_{i=0}^{n} c_i(G) \cdot X^i.$$

It is well known that $n = |V(G)|$, $-c_2(G) = |E(G)|$, and $-c_3(G)$ equals twice the number of triangles of G. The second largest zero $\lambda_2(G)$ of $P(G;X)$ gives a lower bound to the conductivity of G, cf. [GR01].

The Tutte polynomial of G is defined as

$$T(G;X,Y) = \sum_{F \subseteq E(G)} (X-1)^{r\langle E\rangle - r\langle F\rangle}(Y-1)^{n\langle F\rangle} \tag{3}$$

where $k\langle F\rangle$ is the number of connected components of the spanning subgraph defined by F, $r\langle F\rangle = |V| - k\langle F\rangle$ is its rank and $n\langle F\rangle = |F| - |V| + k\langle F\rangle$ is its nullity.

The Tutte polynomial $T(G;X,Y)$ has remarkable evaluations which count certain configurations of the graph G, cf. [Wel93].

(i) $T(G;1,1)$ is the number of spanning trees of G,
(ii) $T(G;1,2)$ is the number of connected spanning subgraphs of G,
(iii) $T(G;2,1)$ is the number of spanning forest of G,
(iv) $T(G;2,2) = 2^{|E|}$ is the number of spanning subgraphs of G,
(v) For connected graphs, $T(G;1-k,0)$ is the number of proper k-vertex colorings of G,
(vi) For connected graphs, $T(G;2,0)$ is the number of acyclic orientations of G,
(vii) $T(G;0,-2)$ is the number of Eulerian orientations of G.

All these are also graph parameters which take values in $\mathbb{N}$. More sophisticated evaluations of the Tutte polynomial can be found in [Goo06, Goo08].

For now it suffices to know that the Tutte polynomial, the matching polynomial, the characteristic polynomial, all discussed in [GR01, Mak07], and the interlace polynomial, defined in [ABS04, Cou], and virtually all the prominent graph polynomials in the literature, are *MSOL*-definable τ_3-polynomials, independently of the order $<_E$. Furthermore, the operations $\sqcup_k$ are all *MSOL*-smooth on τ_3-structures for order-invariant sentences.

The following is a consequence of the Finite Rank Theorem (Theorem 9):

Theorem 12 ([GKM08]). *Let f be a τ-invariant and $\Box$ be an MSOL-compatible operation on τ-structures. If $r(f, \Box)$ is infinite, then f is not an evaluation of an MSOL-definable τ-polynomial.*

In [GKM08] many examples are given for graph parameters. This includes all $\sqcup$-maximizing (minimizing) graph parameters, such as the clique number $\omega(G)$, the chromatic number $\chi(G)$. An example with infinite rank for $\sqcup$ which is not $\sqcup$-maximizinga is the average degree of a graph. There one notes that the connection matrix contains a Cauchy matrix as a submatrix.

4.3 More Graph Polynomials Which Are Not *MSOL*-Definable

Harmonious and complete colorings. Complete colorings, also called achromatic colorings, were introduced in [HHP67]. Harmonious colorings were introduced in [HK83]. For surveys, cf. [Edw97, HM97].

Definition 1. *(i) A proper vertex coloring is* harmonious*, if each pair of colors appears at most once along an edge. We denote by $\chi_{harm}(G)$ the least k such that G has a harmonious proper k-coloring.*

(ii) A proper vertex coloring is complete*, if each pair of colors appears at least once along an edge. We denote by $\chi_{comp}(G)$ the largest k such that G has a complete proper k-coloring.*

(iii) Let $\chi_{harm}(G; k)$ and $\chi_{comp}(G; k)$ denote the number of harmonious, respectively complete proper k-colorings of G.

Proposition 13. *(i) $\chi_{harm}(G; k)$ is a polynomial in k.*
(ii) $\chi_{comp}(G; k)$ is not a polynomial in k.

Proof. (i) follows from [MZ06], but it is not difficult to prove it directly.
(ii) $\chi_{comp}(G; k) = 0$ for large enough k. $\Box$

Theorem 14. *$\chi_{harm}(G)$ and $\chi_{comp}(G)$ are graph parameters which are not evaluations of order invariant MSOL-definable graph polynomials over tau_3.*

Proof. $\chi_{comp}(G)$ is maximizing, so we can apply Proposition 2.

For $\chi_{harm}(G)$ we observe that, for stars S_n, a set of n edges which meet all in one single vertex, we have

$$\chi_{harm}(S_n \sqcup S_m) = \max\{\chi_{harm}(S_m), \chi_{harm}(S_n)\} + 1.$$

Now the argument proceeds like in the case of a maximizing graph parameter.

Theorem 15. *$\chi_{harm}(G; k)$ is not an order invariant MSOL-definable graph polynomial over τ_3.*

Proof. Let L_i denote the graph which consists of i vertex disjoint edges. We look at $M(\chi_{harm}(G, k), \sqcup)$ restricted to the graphs $L_i, i \in \mathbb{N}$, which we denote by $M_L(k)$ and its rank by $r_L(k)$. We note that $\chi_{harm}(L_i \sqcup L_j) = 0$ iff $i + j > \binom{k}{2}$. Therefore, $r_L(k) = \binom{k}{2}$ which is not bounded, contradicting Theorem 9.

Remark 1. *It is shown in [EM95], that computing $\chi_{harm}(G)$ is* **NP***-complete already for trees. This, together with the fact, proven in [Mak05], that evaluations of invariantly CMSOL-definable graph polynomials are polynomial time for graphs of tree-width at most k, shows that $\chi_{harm}(G; X)$ is not invariantly CMSOL-definable, unless* $\mathbf{P} = \mathbf{NP}$. *Our proof above eliminates the complexity theoretic hypothesis* $\mathbf{P} = \mathbf{NP}$.

Convex colorings. A vertex coloring of a graph $G = (V, E)$ with k colors ($k \in \mathbb{N}$) is a function $f : V \to [k]$. f is *convex* if for every $i \in [k]$ the colorclass $f^{-1}(i)$ induces a connected subgraph. For a partial function $f_0 : V \to [k]$ we say that f_0 is convex if there a is a total function f extending f_0 which is convex. In this case we also say that f is a *convex extension* of f_0. Convex extensions of partial colorings of trees have been introduced in the context of phylogenetic trees by S. Moran and S. Snir [MS07].

The existence problem of convex colorings for an arbitrary graph G is easily solved by trying to color every connected component by one color, and only depends on the number of colors available and the number of connected components of G. It follows from [MZ06, KMZ08] that the number of convex colorings of a graph G is a polynomial in k, which we denote by $conv(G, k)$. For $k = 1$ we have $conv(G, 1) = 1$, if G is connected, and $conv(G, 1) = 0$ otherwise. It has been shown by S. Noble and A. Goodall[2] that computing $conv(G, 2)$ is $\sharp\mathbf{P}$-hard. It follows, using a similar argument as in [Lin86], that computing $conv(G, k)$ is $\sharp\mathbf{P}$-hard for every $k \in \mathbb{N} - \{0, 1\}$. On the graphs E_n convex colorings have to color every vertex with a different color. It follows again that $r(conv(G, k), \sqcup)$ tends to infinity with k, and we get

Proposition 16. *The graph polynomial $conv(G, k)$ is not MSOL-definable.*

5 Open Problems

We have discussed various aspects of connection matrices of graph parameters introduced in [FLS07], and have generalized them for τ-invariants and various binary operations between τ-structures. We have shown that the rank of connection matrices is finite for *MSOL*-definable τ-invariants and *MSOL*-compatible binary operations between τ-structures. We used this to show that various graph parameters and graph polynomials are not *MSOL*-definable.

[2] Personal communication.

In the case of partition functions knowing the exact rank $r(f, \sqcup_k)$ allows us to compute f on graphs of tree-width at most k in polynomial time with improved constants on the running time. Can this be generalized?

This leads us the the following questions about τ-invariants in general, although we formulate them for graphs..

Open Problem 1. *Assume $M(f, \Box)$ has rank r and an $(r \times r)$-submatrix M_r of maximal rank is given. Under what conditions on $\Box$ can we compute all the entries of $M(f, \Box)$ from M_r and the computability of $\Box$? What is the complexity of computing the entry $f(G_i \Box G_j)$ of $M(f, \Box)$?*

Open Problem 2. *Under what conditions on the graph parameter f and on $\Box$ can we compute the rank $r(f, \Box)$ precisely, or at least give reasonable lower and upper bounds?*

Open Problem 3. *Let f be a graph parameter on k-graphs and let $r(f, \sqcup_j)$ be finite for every $j \leq k$. Is it true that f can be computed in polynomial time on graphs of tree-width at most k.*

Open Problem 4. *In case the Open Problem 3 has a positive answer, is there an analogue for clique-width?*

I am pretty convinced that the answer are positive. In order to attack the Open Problems above it may be useful to look at connection matrices restricted to a graph property Φ and an operation $\Box$ such that

(i) $\Box$ preserves Φ, i.e., if $G_1 \in \Phi$ and $G_2 \in \Phi$ then also $G_1 \Box G_2 \in \Phi$, and
(ii) the size of $G_1 \Box G_2$ is bigger than the size of G_1 and G_2, for example

$$|V(G_1 \Box G_2)| \geq |V(G_1)| + |V(G_2)|$$

(iii) For every graph $G \in \Phi$ we can effectively find non-trivial G_1 and G_2 such that $G = G_1 \Box G_2$.

Examples for Φ and $\Box$ satisfying these conditions are trees with root a with an additional distinguished node b and $\sqcup_1$ identifying a from one tree with b from the other. Another example are the cliques with the join operation $\bowtie$, or graphs with no edges and the disjoint union $\sqcup$.

Acknowledgments

I wish to thank my co-authors B. Godlin and T. Kotek for their contributions in [GKM08], to T. Kotek who allowed me to include his Proposition 10 from his thesis work, to L. Lyaudet for proof reading these notes, and to L. Lovász and A. Schrijver for their interest and encouragement. Finally I would like to thank R. Ramanujam and S. Sarukkai for inviting me to the 3rd Indian Conference on Logic and its Applications (ICLA) and their insistence that I write up the outline of my invited lecture.

References

[ABS04] Arratia, R., Bollobás, B., Sorkin, G.B.: The interlace polynomial of a graph. Journal of Combinatorial Theory, Series B 92, 199–233 (2004)

[CMR01] Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second order logic. Discrete Applied Mathematics 108(1-2), 23–52 (2001)

[CO00] Courcelle, B., Olariu, S.: Upper bounds to the clique–width of graphs. Discrete Applied Mathematics 101, 77–114 (2000)

[Cou] Courcelle, B.: A multivariate interlace polynomial (preprint) (December 2006)

[Die96] Diestel, R.: Graph Theory. Graduate Texts in Mathematics. Springer, Heidelberg (1996)

[dR84] de Rougemont, M.: Uniform definability on finite structures with successor. In: SIGACT 1984, pp. 409–417 (1984)

[Edw97] Edwards, K.: The harmonious chromatic number and the achromatic number. In: Bailey, R.A. (ed.) Survey in Combinatorics. London Math. Soc. Lecture Note Ser., vol. 241, pp. 13–47. Cambridge Univ. Press, Cambridge (1997)

[EM95] Edwards, K., McDiarmid, C.: The complexity of harmonious colouring for trees. Discrete Appl. Math. 57(2-3), 133–144 (1995)

[FLS07] Freedman, M., Lovász, L., Schrijver, A.: Reflection positivity, rank connectivity, and homomorphisms of graphs. Journal of AMS 20, 37–51 (2007)

[GKM08] Godlin, B., Kotek, T., Makowsky, J.A.: Evaluation of graph polynomials. In: International Symposium on Theoretical Programming. LNCS. Springer, Heidelberg (2008)

[Goo06] Goodall, A.J.: Some new evaluations of the Tutte polynomial. Journal of Combinatorial Theory, Series B 96, 207–224 (2006)

[Goo08] Goodall, A.J.: Parity, eulerian subgraphs and the Tutte polynomial. Journal of Combinatorial Theory, Series B 98(3), 599–628 (2008)

[GR01] Godsil, C., Royle, G.: Algebraic Graph Theory. Graduate Texts in Mathematics. Springer, Heidelberg (2001)

[HHP67] Harary, F., Hedetniemi, S., Prins, G.: An interpolation theorem for graphical homomorphisms. Portugal. Math. 26, 453–462 (1967)

[HK83] Hopcroft, J.E., Krishnamoorthy, M.S.: On the harmonious coloring of graphs. SIAM J. Algebraic Discrete Methods 4, 306–311 (1983)

[HM97] Hughes, F., MacGillivray, G.: The achromatic number of graphs: a survey and some new results. Bull. Inst. Combin. Appl. 19, 27–56 (1997)

[KMZ08] Kotek, T., Makowsky, J.A., Zilber, B.: On counting generalized colorings. In: Kaminski, M., Martini, S. (eds.) CSL 2008. LNCS, vol. 5213, pp. 339–353. Springer, Heidelberg (2008)

[Lin86] Linial, M.: Hard enumeration problems in geometry and combinatorics. SIAM Journal of Algebraic and Discrete Methods 7, 331–335 (1986)

[LMST07] Linial, N., Matousek, J., Sheffet, O., Tardos, G.: Graph coloring with no large monochromatic components. arXiv:math/0703362 (2007)

[Lov07] Lovász, L.: Connection matrics. In: Grimmet, G., McDiarmid, C. (eds.) Combinatorics, Complexity and Chance, A Tribute to Dominic Welsh, pp. 179–190. Oxford University Press, Oxford (2007)

[Mak04] Makowsky, J.A.: Algorithmic uses of the Feferman-Vaught theorem. Annals of Pure and Applied Logic 126(1-3), 159–213 (2004)

[Mak05] Makowsky, J.A.: Colored Tutte polynomials and Kauffman brackets on graphs of bounded tree width. Disc. Appl. Math. 145(2), 276–290 (2005)

[Mak07] Makowsky, J.A.: From a zoo to a zoology: Towards a general theory of graph polynomials. Theory of Computing Systems (July 2007), `http://dx.doi.org/10.1017/s00224--007--9022--9`

[MS07] Moran, S., Snir, S.: Efficient approximation of convex recolorings. Journal of Computer and System Sciences 73(7), 1078–1089 (2007)

[MZ06] Makowsky, J.A., Zilber, B.: Polynomial invariants of graphs and totally categorical theories. MODNET Preprint No. 21 (2006), `http://www.logique.jussieu.fr/modnet/Publications/Preprint%20server`

[Oum05] Oum, S.: Approximating rank-width and clique-width quickly. In: Kratsch, D. (ed.) WG 2005. LNCS, vol. 3787, pp. 49–58. Springer, Heidelberg (2005)

[Sch08] Schrijver, A.: Polynomial and tensor invariants and combinatorial parameters. Amsterdam (preprint, 2008)

[Sze07] Szegedy, B.: Edge coloring models and reflection positivity, arXiv: math.CO/0505035 (2007)

[Wel93] Welsh, D.J.A.: Complexity: Knots, Colourings and Counting. London Mathematical Society Lecture Notes Series, vol. 186. Cambridge University Press, Cambridge (1993)

Knowledge, Games and Tales from the East

Rohit Parikh

Brooklyn College of CUNY and CUNY Graduate Center

Abstract. We introduce some basic concepts from Game Theory and related areas and show how various insights for which we thank game theory have already occurred in the past in some tales from ancient literature, both Indian and otherwise.

1 Games

We shall usually talk about two player games. The players are typically called **Row** and **Column**, but more catchy names may arise in specific contexts.

In so called normal form games, each player has a finite set of **strategies**, call them S_1 and S_2, and each can choose a particular strategy from their own set. Once the players have chosen their strategies, there are payoffs which depend on *both* the strategies. So suppose that player Row chooses strategy a and Column chooses strategy b, then the payoffs would be $p_r(a, b)$ and $p_c(a, b)$. We may also refer to Row and Column as players 1 and 2 respectively.

Suppose Row has chosen a and Column has chosen b, then (a, b) constitutes a **Nash equilibrium** if, *given that column is playing* b, Row has nothing better than a, and given that Row is playing a, Column has nothing better than b. In other words $p_r(a, b) \geq p_r(a', b)$ for all a' and $p_c(a, b) \geq p_c(a, b')$ for all b'.

Given two strategies a, a' for Row, we say that a is **dominated** by a' if regardless of what Column plays, a' always gives a better outcome for Row. Thus $p_r(a, b) \leq p_r(a', b)$ for all b and $p_r(a, b) < p_r(a', b)$ for at least one b. Sinilarly for dominance of a Column strategy b by b'. It is normally accepted that a player would never play a dominated strategy, and the opponent may then make his plans based on this fact.

We now give examples of various games in the literature.

1.1 Battle of the Sexes

In this game, the wife (Row) wants to go to the Opera and the husband (Column) wants to watch football. But each would rather go together than watch their favourite thing by themselves. So here are the payoffs. Row's payoffs in each box are listed first.

	Opera	Footb
Opera	$2, 1$	$0, 0$
Footb	$0, 0$	$1, 2$

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 65–76, 2009.

If they go to different events, they are not happy so the payoffs are zero for both. If they go to the *same event*, then both have positive payoffs, but the wife's is higher if they go to the Opera and the husband's is higher if they go to football. There are two Nash equilibria, the NW one which is (2,1), and the SE one which is (1,2).

The fact that (1,2) is a Nash equilibrium can be seen geometrically. Row can change the row, but if she does her payoff will move from 1 to 0, and she will be worse off. Similarly, Column can change the column, but if he does, his payoff will change from 2 to 0, and he will be worse off.

1.2 Chicken

In this rather dangerous game, two cars race towards each other. If one goes straight and the other swerves, then the one who swerves has shown fear, and is called *chicken*. He is embarrassed while the other crows. If neither swerves then there is an accident which they both regret – if they survive.

	Swerve	Straight
Swerve	$4, 4$	$2, 7$
Straight	$7, 2$	$-10, -10$

There are two Nash equilibria, the NE one which is (2,7), with Row being the 'chicken' and the SW one which is (7,2) wth Column in that role.

1.3 Matching Pennies

In this game, *Row* is the matcher and *Column* is the mismatcher. Both parties exhibit a penny and if both pennies match (are both showing heads or both showing tails) then Row wins. If one is showing heads and the other tails (mismatch), then Column wins. There are *no* Nash equilibria in this game (there *is* a mixed strategy equilibrium, but we shall not consider those here).

	Heads	Tails
Heads	$1, -1$	$-1, 1$
Tails	$-1, 1$	$1, -1$

1.4 Prisoner's Dilemma

In this game, two men are arrested and invited to testify against each other. If neither testifies, then there is a small penalty since there is no real evidence. But if one *defects* (testifies) and the other does not, then the defecter goes free and the other gets a large sentence. If both defect they both get medium sentences. Jointly they are better off (The payoffs are 2 each) if neither defects, but for both of them, defecting is the dominant strategy and they end up with (1,1) which is worse.

	Coop	Def
Coop	$2, 2$	$0, 3$
Def	$3, 0$	$1, 1$

There is a unique, rather bad Nash equilibrium at SE with (1,1), while the (2,2) solution on NW, though better for *both*, is not a Nash equilibrium.

We now discuss the first one of our folk examples. We start with an actual example from the Economics literature and then relate it to a story from Indian history.

2 Tales from the East

2.1 Tragedy of the Commons

From **"The Tragedy of the Commons"** by Garrett Hardin, 1968.

The tragedy of the commons develops in this way. Picture a pasture open to all. It is to be expected that each herdsman will try to keep as many cattle as possible on the commons. Such an arrangement may work reasonably satisfactorily for centuries because tribal wars, poaching, and disease keep the numbers of both man and beast well below the carrying capacity of the land. Finally, however, comes the day of reckoning, that is, the day when the long-desired goal of social stability becomes a reality. At this point, the inherent logic of the commons remorselessly generates tragedy.

As a rational being, each herdsman seeks to maximize his gain. Explicitly or implicitly, more or less consciously, he asks, "What is the utility to me of adding one more animal to my herd?" This utility has one negative and one positive component.

1. The positive component is a function of the increment of one animal. Since the herdsman receives all the proceeds from the sale of the additional animal, the positive utility is nearly +1.

2. The negative component is a function of the additional overgrazing created by one more animal. Since, however, the effects of overgrazing are shared by all the herdsmen, the negative utility for any particular decisionmaking herdsman is only a fraction of -1.

Adding together the component partial utilities, the rational herdsman concludes that the only sensible course for him to pursue is to add another animal to his herd. And another.... But this is the conclusion reached by each and every rational herdsman sharing a commons. Therein is the tragedy. Each man is locked into a system that compels him to increase his herd without limit – in a world that is limited. Ruin is the destination toward which all men rush, each pursuing his own best interest in a society that believes in the freedom of the commons. Freedom in a commons brings ruin to all.

From "The Tragedy of the Commons" by Garrett Hardin, [4]. But Hardin was anticipated in India by four hundred years! The following is from the famous **Akbar Birbal** collection of stories. Akbar was the third Mughal emperor and the grandfather of Shah Jehan who built the Taj Mahal as a monument (and mausoleum) for his wife. Birbal was one of his ministers and well known (at least in stories) for his wit and intelligence. Both lived in the second half of the sixteenth century.

2.2 Birbal Story

One day Akbar Badshah said something to Birbal and asked for an answer. Birbal gave the very same reply that was in the king's own mind. Hearing this, the king said, *This is just what I was thinking also.* Birbal said, *Lord and Guide, this is a case of* a hundred wise men, one opinion *(in Hindi, sau siyane ek mat)*. The king said, *This proverb is indeed well-known.* Then Birbal petitioned, *Refuge of the World, if you are so inclined, please test this matter.* The king replied, *Very good.* The moment he heard this, Birbal sent for a hundred wise men from the city. And the men came into the king's presence that night. Showing them an empty well, Birbal said, *His Majesty orders that at once every man will bring one bucket full of milk and pour it in this well.* The moment they heard the royal order, every one reflected that *where there were ninety-nine buckets of milk, how could one bucket of water be detected?* Each one brought only water and poured it in. Birbal showed it to the king. The king said to them all, *What were you thinking, to disobey my order? Tell the truth, or I'll treat you harshly!* Every one of them said with folded hands, *Refuge of the World, whether you kill us or spare us, the thought came into this slave's mind that where there were ninety-nine buckets of milk, how could one bucket of water be detected?* Hearing this from the lips of all of them, the king said to Birbal, *What I'd heard with my ears, I've now seen before my eyes:* a hundred wise men, one opinion!

Birbal lived from 1528 to 1586, and died in the battle of Malandari Pass, in Northwest India.

http://en.wikipedia.org/wiki/Akbar_the_Great
http://en.wikipedia.org/wiki/Birbal

Analysis: What is common between the example which Hardin gives and the Akbar-Birbal story? In each case, the individual benefits at the cost of the group. In the Hardin case, the herdsman benefits by having one more animal. In the Birbal case, the "wise man" benefits by saving one pot of milk. In each case the group is harmed. In the case of the herdsmen, the common is overgrazed and the grass dies. In the Akbar-Birbal case, there is a danger that if the cheating is discovered, all hundred men face the threat of prison or even execution. Akbar was a benign king,[1] but not entirely immune to anger. Also, in each case, cheating is a dominant strategy. If most of the others are cheating, it does no extra harm if you cheat too. And if most of the others are not cheating, then again it does no extra harm if you are one of the rare cheaters. But if everyone practices their dominant strategy and cheats, then there can be disaster for the whole group.

2.3 Can We Always Believe What Others Tell Us? Solomon Story

The following story is from the *Old Testament*, first book of *Kings*, chapter 3. Then came there two women, that were harlots, unto the king, and stood before

[1] Akbar, though a Muslim, worked hard to create amity between Hindus and Muslims, even marrying a Hindu wife, and having endless discussions on religion with Hindus, Christians and Jains.

him. And the one woman said, *O my lord, I and this woman dwell in one house; and I was delivered of a child with her in the house. And it came to pass the third day after that I was delivered, that this woman was delivered also: and we were together; there was no stranger with us in the house, save we two in the house. And this woman's child died in the night; because she overlaid it. And she arose at midnight, and took my son from beside me, while thine handmaid slept, and laid it in her bosom, and laid her dead child in my bosom.*

And when I rose in the morning to give my child suck, behold, it was dead: but when I had considered it in the morning, behold, it was not my son, which I did bear. And the other woman said, *Nay; but the living is my son, and the dead is thy son.* And this said, *No; but the dead is thy son, and the living is my son.* Thus they spake before the king.

Then said the king, *The one saith, This is my son that liveth, and thy son is the dead: and the other saith, Nay; but thy son is the dead, and my son is the living.*

And the king said, *Bring me a sword.* And they brought a sword before the king.

And the king said, *Divide the living child in two, and give half to the one, and half to the other.* Then spake the woman whose the living child was unto the king, for her bowels yearned upon her son, and she said, *O my lord, give her the living child, and in no wise slay it.* But the other said, *Let it be neither mine nor thine, but divide it.*

Then the king answered and said, *Give her the living child, and in no wise slay it: she is the mother thereof.*

Analysis: Let **M** stand for "I get the child", **O** stand for "The other woman gets the child", and **K** stand for "The child is killed.

Both women prefer **M** to **O**. However, Solomon relies on the fact that the real mother prefers **O** to **K** whereas the non-mother prefers **K** to **O**. Thus the orderings are: $M > O > K$ for the real mother and $M > K > O$ for the non-mother. Asked to choose between O and K, the real mother chooses O and the non-mother chooses K. This enables Solomon to discover the real mother. Solomon is trying to implement what is called the *revelation principle* according to which people reveal their real opinions by how they act. However, Solomon's strategy has a bug. If the non-mother knows what his plans are, all she has to do is to say, "Oh, I too would rather the other woman took the child than have it killed." And then Solomon would be in a quandary.

Such a behavior would be an example of what is called *strategizing* [3,7,8], where you express a preference different from your actual one in order to get a better result. There *is*, however, a solution which depends on money, or let us say, public service. Suppose the real mother is willing to do three months public service to get the child, but the non-mother is only willing to do one month. Solomon of course does not know which is which but he can use this information and the following procedure to discover who is the real mother. Thus here is the plan. Suppose the two women are Anna and Beth. Solomon first asks Anna, *Is the child yours?* If Anna says no, Beth gets the child and that ends the matter.

If Anna says, *It is my child*, then Beth is asked *Is the child yours?* If Beth says no, Anna gets the child and that ends the matter.

If Beth also says, *It is my child*, then Beth gets the child, and does two months public service. Anna also does one week's public service.

It is easy to see that only the real mother will say, *It is my child*, and no public service needs to be performed.

For suppose that Anna is the real mother. She can safely say, *It i s my child* because when Beth is asked next, she does not want to do two months service to get the child. Anna will get the child without any problem. I leave it to you to work out what happens if Beth is the real mother. For a recent paper on such problems, see

http://ideas.repec.org/p/pra/mprapa/8801.html

It might have struck the reader that while the outcome is fair to both women, the algorithm is not symmetric. But there do exist symmetric algorithms based on the idea of the Vickrey auction [5], and the one in the paper cited just above is an example.

2.4 Cheap Talk

The following examples are slightly adapted from [2].

Laxmi is applying to Rayco for a job, and Rayco asks if her ability is high or low.

Will Laxmi speak the truth, and can Rayco trust her?

		Rayco	
		High	**Low**
Laxmi	**High**	(3,3)	(0,0)
	Low	(0,0)	(2,2)

Fig. 1.

In the scenario above, Figure 1, Rayco prefers to hire Laxmi for the high position if she has high ability and the low position if her ability is low. If they ask her about her ability, Laxmi has nothing to gain by lying about her

qualifications and Rayco can trust her. In particular, if her ability is low and she lies that it is high, Rayco would give her the higher position and she would be frustrated so that the higher salary would not be an advantage.

But suppose instead (Figure 2) that Laxmi feels she can get away with having a better job even with worse ability. Perhaps she feels she can 'wing it', or pass on her more difficult work to others. If Laxmi's ability is low, she still prefers the higher paying job so she would like to entice Rayco (which chooses the job she is offered) into the bottom left box. But if Rayco knows her payoffs, they will be careful not to believe her bare statement that she has high ability.

		Rayco	
		High	**Low**
Laxmi	**High**	(3,3)	(0,0)
	Low	(3,0)	(2,2)

Fig. 2.

In this scenario, Laxmi *can* profit from having a high job even if her ability is low, her payoff is 3 in any case. So Laxmi has nothing to lose by lying about her qualifications and Rayco cannot trust her.

The moral is, as we all know, *If someone tells us something, then before believing it, ask if they could gain by lying.* There is a bit more to cheap talk than this but we shall not go into details.

2.5 The Mahabharata

The Kurukshetra War forms an essential component of the Hindu epic Mahabharata. According to Mahabharata, a dynastic struggle between sibling clans of Kauravas and the Pandavas for the throne of Hastinapura resulted in a battle in which a number of ancient kingdoms participated as allies of the rival clans. The location of the battle was Kurukshetra in the modern state of Haryana in India. Mahabharata states that the war lasted eighteen days during which vast armies from all over ancient India fought alongside the two rivals. Despite only

referring to these eighteen days, the war narrative forms more than a quarter of the book, suggesting its relative importance within the book.

http://en.wikipedia.org/wiki/Kurukshetra_War

During the war, Drona, who was the teacher of both Pandavas and Kauravas, and an expert bowman, is fighting on the side of the Kauravas and the Pandavas are desperate as they do not know what to do! Luckily, Drona's son is called *Ashwathama* as is an elephant owned by the Pandavas.

Yudhisthira was the oldest of the five Pandava brothers, and had a reputation for honesty. His role in what happens is crucial. On the 15th day of the war. Krishna asked Yudhisthira to proclaim that Drona's son Ashwathama has died, so that the invincible and destructive Kuru commander would give up his arms and thus could be killed. Bhima proceeds to kill an elephant named Ashwathama, and loudly proclaims that Ashwathama is dead.

Drona knows that only Yudhisthira, with his firm adherence to the truth, could tell him for sure if his son had died. When Drona approaches Yudhisthira to seek to confirm this, Yudhisthira tells him that Ashwathama is dead..., then, ..the elephant, but this last part is drowned out by the sound of trumpets and conchshells being sounded as if in triumph, on Krishna's instruction. Yudhisthira cannot make himself tell a lie, despite the fact that if Drona continued to fight, the Pandavas and the cause of dharma itself would lose. When he speaks his half-lie, Yudhisthira's feet and chariot descend to the ground momentarily. Drona is disheartened, and lays down his weapons. He is then killed by Dhristadyumna.

It is said that Drona's soul, by meditation had already left his body before Dhristadyumna could strike. His death greatly saddens Arjuna, who had hoped to capture him alive.

http://en.wikipedia.org/wiki/Drona

Clearly the Pandavas had an incentive to lie (as Laxmi does in our second example with Rayco), but Drona assumed that in the case of Yudhisthira, the loyalty to truth would override his self-interest. It so turned out that Drona was only partly right.

2.6 The Two Horsemen

Suppose we want to find out which of two horses is faster. This is easy, we race them against each other. The horse which reaches the goal first is the faster horse. And surely this method should also tell us which horse is *slower*, it is the other one. However, there is a complication which will be instructive.

Two horsemen are on a forest path chatting about something. A passerby M, the mischief maker, comes along and having plenty of time and a desire for amusement, suggests that they race against each other to a tree a short distance away and he will give a prize of \$100. However, there is an interesting twist. He will give the \$100 to the owner of the *slower* horse. Let us call the two horsemen Bill and Joe. Joe's horse can go at 35 miles per hour, whereas Bill's horse can only go 30 miles per hour. Since Bill has the slower horse, he should get the

\$100. The two horsemen start, but soon realize that there is a problem. Each one is trying to go slower than the other and it is obvious that the race is not going to finish. There is a broad smile on the canny passerby's face as he sees that he is having some amusement at no cost.

Figure 3, below, explains the difficulty. Here Bill is the row player and Joe is the column player. Each horseman can make his horse go at any speed upto its maximum. But he has no reason to use the maximum. And in figure 3, the left columns are dominant (yield a better payoff) for Joe and the top rows are dominant for Bill. Thus they end up in the top left hand corner, with both horses going at 0 miles per hour.

	0	10	20	30	35
0	0, 0	100, 0	100, 0	100, 0	100, 0
10	0, 100	0, 0	100, 0	100, 0	100, 0
20	0, 100	0, 100	0, 0	100, 0	100, 0
30	0, 100	0, 100	0, 100	0, 0	100, 0

Fig. 3.

However, along comes another passerby, let us call her S, the problem solver, and the situation is explained to her. She turns out to have a clever solution. She advises the two men to switch horses. Now each man has an incentive to go fast, because by making his competitor's horse go faster, he is helping his own horse to 'win'! Figure 4 shows how the dominant strategies have changed. Now Joe (playing row) is better off to the bottom, and Bill playing column is better off to the right – they are both urging the horse they are riding (their opponent's horse) as fast as the horse can go. Thus they end up in the bottom right corner of figure 4. Joe's horse, ridden by Bill comes first and Bill gets the \$100 as he should.

Of course, if the first passerby had really *only* wanted to reward the slower horse (or its owner) he could have done this without the horses being switched and for a little extra money. He could have kept quiet about the \$100 and offered

	0	10	20	30	35
0	0, 0	0, 100	0, 100	0, 100	0, 100
10	100, 0	0, 0	0, 100	0, 100	0, 100
20	100, 0	100, 0	0, 0	0, 100	0, 100
30	100, 0	100, 0	100, 0	0, 0	0, 100

Fig. 4.

a prize of $10 to the owner of the faster horse. Then when the race was over, he would hand over the $10 to Joe and $100 to Bill. Here the effect would be achieved by hiding from the two horsemen what their best strategy was, and to fool them into thinking that some other action was in fact better. While the problem of finding the faster horse, and that of finding the slower, are equivalent algorithmically, they are not equivalent game theoretically when the men ride their own horses. The equivalence is restored when the two men switch horses. For a practical analogue of the two horses example, consider the issue of grades and letters of recommendation. Suppose that Prof. Meyer is writing a letter of recommendation for his student Maria and Prof. Shankar is writing one for his student Peter. Both believe that their respective students are good, but only good. Not very good, not excellent, just good. Both also know that only one student can get the job or scholarship. Under this circumstance, it is clear that both of the advisers are best off writing letters saying that their respective student is excellent. This is strategic behaviour in a domain familiar to all of us.

Sometimes employers will try to counter this by appealing to third parties for an evaluation, but the close knowledge that the two advisers have of their advisees cannot be discovered very easily.

In Figure 5 above, J represents job and NJ represents no job for the student. Then Meyer's lower strategies dominate his upper ones. And for Shankar, his rightward strategies dominate the strategies to the left. Hence, with each playing his dominant strategies, they end up in the lower right hand corner with neither student getting the job.

We do assume that in case of a tie neither student is hired. This of course need not be true in reality – perhaps one would be chosen at random. But if one of

Shankar's choices

Meyer's choices

	G	VG	E
G	NJ , NJ	NJ , J	NJ , J
VG	J , NJ	NJ , NJ	NJ , J
E	J , NJ	J , NJ	NJ , NJ

Fig. 5.

the students is actually superior, that information cannot be elicited by asking their advisers. Sometimes the National Science Foundation, giving out grants, tends to ask people to reveal their connections with various referees. Then some semblance of neutrality can be achieved.

2.7 A Bankruptcy Problem

This problem has been studied by Aumann and Maschler [1]. A man dies leaving debts $d_1, ..., d_n$ totalling more than his estate E. How should the estate be divided among the creditors?

Here are some solutions from the *Babylonian Talmud.* In all cases, $n = 3, d_1 = 100, d_2 = 200, d_3 = 300$. Let the amounts actually awarded be x_1, x_2, x_3.

$E = 100$.
The amounts awarded are $x_i = 33.3$ for $i = 1, 2, 3$
$E = 200$. $x_1 = 50, x_2 = 75, x_3 = 75$
$E = 300$. $x_1 = 50, x_2 = 100, x_3 = 150$.

What explains these numbers?

The Contested Garment Principle: Suppose two people A, B are claiming 50 and 90 respectively from a debtor whose total worth is 100. Then A has conceded 50 and B has conceded 10. Then B gets the 50 conceded by A and A gets the 10 conceded by B. That leaves 40 which is equally divided. Thus A gets 30 and B gets 70.

Similarly, if E is a garment, A claims half of it and B claims all, then A ends up with .25 and B with .75 of the garment.

Note that under the contested garment principle the results are monotonic in the claims and also in the total amount available for division.

Definition 2.1. *A* bankrupcy problem *is defined as a pair* $E; d$ *where* $d = (d_1, ..., d_n)$*,* $0 \leq d_1 \leq d_2 \leq ... \leq d_n$ *and* $0 \leq E \leq d_1 + ...d_n$*. A* solution *to such a problem is an* n*-tuple* $x = (x_1, ..., x_n)$ *of real numbers with*

$$x_1 + x_2 + ... + x_n = E$$

A solution is called CG-consistent *if for all* $i \neq j$*, the division of* $x_i + x_j$ *prescribed by the contested garment principle for claims* d_i, d_j *is* (x_i, x_j)*.*

Theorem 2.2. *(Aumann, Maschler) Each bankrupcy problem has a unique consistent solution.*

Proof. (uniqueness) Suppose that x, y are different solutions. Then there must be i, j such that i receives more in the second case and j receives less. Assume wlog that $x_i + x_j \leq y_i + y_j$. Thus we have $x_i < y_i$, $x_j > y_j$ and $x_i + x_j \leq y_i + y_j$. But the monotonicity principle says that since $y_i + y_j$ is *more*, j should receive more in the y case. contradiction.

Acknowledgements

This research was partially supported by an internal research grant from PSC-CUNY. I thank Sonja Amadae and Andreas Witzel for comments.

References

1. Aumann, R., Maschler, M.: Game-theoretic analysis of a bankruptcy problem from the Talmud. J. Economic Theory 36, 195–213 (1985)
2. Farrell, J., Rabin, M.: Cheap Talk. Journal of Economic Perspectives 10, 103–118 (1996)
3. Gibbard, A.: Manipulation of voting schemes: a general result. Econometrica 41(4), 587–601 (1973)
4. Hardin, G.: The Tragedy of the Commons. Science 162(3859), 1243–1248 (1968)
5. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
6. Parikh, R.: Social Software. Synthese 132, 187–211 (2002)
7. Satterthwaite, M.A.: Strategy-proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. Journal of Economic Theory 10, 187–217 (April 1975)
8. Taylor, A.D.: The manipulability of voting systems. The American Mathematical Monthly (April 2002)

A Para Consistent Fuzzy Logic

Esko Turunen

P.O. Box 553,
33101 Tampere, Finland

Abstract. The root of this work is in Belnap's four valued para consistent logic [2]. Based on a related study of Perny and Tsoukias [11], we introduce para consistent Pavelka style fuzzy sentential logic. Restricted to Lukasiewicz t–norm, our approach and the approach in [11] partly overlap; the main difference lies in the interpretation of the logical connectives implication and negation. The essential mathematical tool proved in this paper is a one–one correspondence between evidence couples and evidence matrices that holds in all injective MV–algebras. Evidence couples associate to each formula α two values a and b that can be interpreted as the degrees of pros and cons for α, respectively. Four values t, f, k, u, interpreted as the degrees of truth, falsehood, contradiction and unknowness of α, respectively, can be calculated. In such an approach truth and falsehood are not each others complements. This paper can be seen as a solution to some open problems presented in [11].

Keywords: Mathematical fuzzy logic, para consistent sentential logic, MV–algebra.

1 Introduction

Four possible values associated with a formula α in Belnap's first order para consistent logic [2] are `true`, `false`, `contradictory` and `unknown`: if there is evidence for α and no evidence against α, then α obtains the value `true` and if there is no evidence for α and evidence against α, then α obtains the value `false`. A value `contradictory` corresponds to a situation where there is simultaneously evidence for α and against α and, finally, α is labeled by value `unknown` if there is no evidence for α nor evidence against α. More formally, the values are associated with ordered couples $\langle 1, 0\rangle$, $\langle 0, 1\rangle$, $\langle 1, 1\rangle$ and $\langle 0, 0\rangle$, respectively.

In [12] Tsoukias introduced an extension of Belnap's logic (named DDT) most importantly because the corresponding algebra of Belnap's original logic is not a Boolean algebra, while the extension is. Indeed, in that paper it was introduced and defined the missing connectives in order to obtain a Boolean algebra. Moreove, it was explained why we get such a structure. Among others it was shown that negation, which was reintroduced in [12] in order to recover some well known tautologies in reasoning, is not a complementation.

In [11] and [14], a continuous valued extension of DDT logic is studied. The authors impose reasonable conditions this continuous valued extension should

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 77–88, 2009.

obey and, after a careful analysis, they come to the conclusion that the graded values are to be computed via

$$t(\alpha) = \min\{B(\alpha), 1 - B(\neg\alpha)\}, \tag{1}$$
$$k(\alpha) = \max\{B(\alpha) + B(\neg\alpha) - 1, 0\}, \tag{2}$$
$$u(\alpha) = \max\{1 - B(\alpha) - B(\neg\alpha), 0\}, \tag{3}$$
$$f(\alpha) = \min\{1 - B(\alpha), B(\neg\alpha)\}. \tag{4}$$

where an ordered couple $\langle B(\alpha), B(\neg\alpha)\rangle$ is given. The intuitive meaning of $B(\alpha)$ and $B(\neg\alpha)$ is the degree of evidence for α and against α, respectively. Moreover, the set of 2×2 matrices of a form

$$\begin{bmatrix} f(\alpha) & k(\alpha) \\ u(\alpha) & t(\alpha) \end{bmatrix}$$

is denoted by $\mathcal{M}$. In [11] it is shown how such a fuzzy version of Belnap's logic can be applied in preference modeling, however, the following open problems is posed:

– the experimentation of different families of De Morgan triples;
– a complete truth calculus for logics conceived as fuzzy extensions of four valued para consistent logics;
– a more thorough investigation of valued sets and valued relations (when the valuation domain is $\mathcal{M}$) and their potential use in the context of preference modeling.

In this paper we accept the challenge to answer some of these problems. Our basic observation is that the algebraic operations in (1) – (4) are expressible by the Lukasiewicz t–norm and the corresponding residuum, i.e. in the Lukasiewicz structure, which is an example of an injective MV–algebra. In [13] it is proved that Pavelka style fuzzy sentential logic is a complete logic in a sense that if the truth value set $\mathbf{L}$ forms an injective MV–algebra, then the set of a–tautologies and the set of a–provable formulae coincide for all $a \in \mathbf{L}$. We therefore consider the problem that, given a truth value set which is an injective MV–algebra, is it possible to transfer an injective MV–structure to the set $\mathcal{M}$, too. The answer turns out to be affirmative, consequently, the corresponding para consistent sentential logic is essentially Pavelka style fuzzy logic. Thus, a rich semantics and syntax is available. For example, Lukasiewicz tautologies as well as Intuitionistic tautologies can be expressed in the framework of this logic. This follows by the fact that we have two sorts of logical connectives conjunction, disjunction, implication and negation interpreted either by the monoidal operations $\bigodot, \bigoplus, \longrightarrow, ^*$ or by the lattice operations $\wedge, \vee, \Rightarrow, ^\star$, respectively (however, neither * nor * is a lattice complementation). Besides, there are many other logical connectives available.

Arieli and Avron [1] developed a logical system based on a class of bilattices (cf. [5]), called logical bilattices, and provided a Gentzen–style calculus for it. This logic is essentially an extension of Belnaps four–valued logic to the standard

language of bilattices, but differs from it for some interesting properties. However, our approach differs from that of Arieli and Avron [1].

Quite recently Dubois [4] published a critical philosophy of science orientated study on Belnap's approach. According to Dubois, the main difficulty lies in the confusion between truth–values and information states. We study para consistent logic from a purely formal point of view without any philosophical contentions. Possible applications of our approach are discussed at the end of the paper.

2 Algebraic Preliminaries

We start by recalling some basic definitions and properties of MV–algebras; all detail can be found in [9, 13]. We also prove some new results that we will utilize later. An *MV-algebra* $\mathbf{L} = \langle L, \oplus, ^*, \mathbf{0} \rangle$ is a structure such that $\langle L, \oplus, \mathbf{0} \rangle$ is a commutative monoid, i.e.,

$$x \oplus y = y \oplus x, \tag{5}$$
$$x \oplus (y \oplus z) = (x \oplus y) \oplus z, \tag{6}$$
$$x \oplus \mathbf{0} = x \tag{7}$$

holds for all elements $x, y, z \in L$ and, moreover,

$$x^{**} = x, \tag{8}$$
$$x \oplus \mathbf{0}^* = \mathbf{0}^*, \tag{9}$$
$$(x^* \oplus y)^* \oplus y = (y^* \oplus x)^* \oplus x. \tag{10}$$

Denote $x \odot y = (x^* \oplus y^*)^*$ and $\mathbf{1} = \mathbf{0}^*$. Then $\langle L, \odot, \mathbf{1} \rangle$ is another commutative monoid and hence

$$x \odot y = y \odot x, \tag{11}$$
$$x \odot (y \odot z) = (x \odot y) \odot z, \tag{12}$$
$$x \odot \mathbf{1} = x \tag{13}$$

holds for all elements $x, y, z \in L$. It is obvious that $x \oplus y = (x^* \odot y^*)^*$, thus the triple $\langle \oplus, ^*, \odot \rangle$ satisfies De Morgan laws. A partial order on the set L is introduced by

$$x \leq y \text{ iff } x^* \oplus y = \mathbf{1} \text{ iff } x \odot y^* = \mathbf{0}. \tag{14}$$

By setting

$$x \vee y = (x^* \oplus y)^* \oplus y, \tag{15}$$
$$x \wedge y = (x^* \vee y^*)^* [= (x^* \odot y)^* \odot y] \tag{16}$$

for all $x, y, z \in L$ the structure $\langle L, \wedge, \vee \rangle$ is a lattice. Moreover, $x \vee y = (x^* \wedge y^*)^*$ holds and therefore the triple $\langle \wedge, ^*, \vee \rangle$, too, satisfies De Morgan laws. However,

the unary operation * called *complementation* is not a lattice complementation. By stipulating

$$x \to y = x^* \oplus y \tag{17}$$

the structure $\langle L, \leq \wedge, \vee, \odot, \to, \mathbf{0}, \mathbf{1}\rangle$ is a residuated lattice with the bottom and top elements $\mathbf{0}, \mathbf{1}$, respectively. In particular, a Galois connection

$$x \odot y \leq z \text{ iff } x \leq y \to z \tag{18}$$

holds for all $x, y, z \in L$. The couple $\langle \odot, \to \rangle$ is an *adjoint couple*. Lattice operations on L can now be expressed via

$$x \vee y = (x \to y) \to y, \tag{19}$$

$$x \wedge y = x \odot (x \to y). \tag{20}$$

A standard example of an MV–algebra is the *Lukasiewicz structure* $\mathcal{L}$: the underlying set is the real unit interval $[0, 1]$ equipped with the usual order and, for each $x, y \in [0, 1]$,

$$x \oplus y = \min\{x + y, 1\}, \tag{21}$$

$$x^* = 1 - x. \tag{22}$$

Moreover,

$$x \odot y = \max\{0, x + y - 1\}, \tag{23}$$

$$x \vee y = \max\{x, y\}, \tag{24}$$

$$x \wedge y = \min\{x, y\}, \tag{25}$$

$$x \to y = \min\{1, 1 - x + y\}, \tag{26}$$

$$x \odot y^* = \max\{x - y, 0\}. \tag{27}$$

For any natural number $m \geq 2$, a finite chain $0 < \frac{1}{m} < \cdots < \frac{m-1}{m} < 1$ can be viewed as an MV–algebra where $\frac{n}{m} \oplus \frac{k}{m} = \min\{\frac{n+k}{m}, 1\}$ and $(\frac{n}{m})^* = \frac{m-n}{m}$. Finally, a structure $\mathcal{L} \cap \mathbf{Q}$ with the Lukasiewicz operations is an example of a countable MV–algebra called *rational Lukasiewicz structure*. All these examples are linear MV–algebras, i.e. the corresponding order is a total order. Moreover, they are MV–subalgebras of the structure $\mathcal{L}$. A Boolean algebra is an MV-algebra such that the monoidal operations $\oplus$, $\odot$ and the lattice operations $\vee$, $\wedge$ coincide, respectively.

An MV–algebra $\mathbf{L}$ is called *complete* if $\bigvee\{a_i |\ i \in \Gamma\}$, $\bigwedge\{a_i |\ i \in \Gamma\} \in L$ for any subset $\{a_i : i \in \Gamma\} \subseteq L$. Complete MV–algebras are completely distributive, that is, they satisfy

$$x \wedge \bigvee_{i \in \Gamma} y_i = \bigvee_{i \in \Gamma} (x \wedge y_i), \quad x \vee \bigwedge_{i \in \Gamma} y_i = \bigwedge_{i \in \Gamma} (x \vee y_i), \tag{28}$$

for any $x \in L$, $\{y_i |\ i \in \Gamma\} \subseteq L$. Thus, in a complete MV–algebra we can define another adjoint couple $\langle \wedge, \Rightarrow \rangle$, where the operation $\Rightarrow$ is defined via

$$x \Rightarrow y = \bigvee\{z |\ x \wedge z \leq y\}. \tag{29}$$

In particular, $x^\star = x \Rightarrow \mathbf{0}$ defines another complementation (called *weak complementation*) in complete MV–algebras. However, weak complementation needs not to be lattice complementation. A *Heyting algebra* H is a bounded lattice such that for all $a, b \in H$ there is a greatest element x in H such that $a \wedge x \leq b$. Thus, to any complete MV–algebra $\langle L, \oplus, ^*, \mathbf{0}\rangle$ there is an associated Heyting algebra $\langle L, \wedge, ^\star, \mathbf{0}, \mathbf{1}\rangle$ with an adjoint couple $\langle \wedge, \Rightarrow\rangle$. The Lukasiewicz structure and all finite MV–algebras are complete as well as the direct product of complete MV–algebras is a complete MV–algebra. However, the rational Lukasiewicz structure is not complete.

A fundamental fact proved by C. C. Chang (cf. [3]) is that any MV–algebra is a subdirect product of Lukasiewicz structures (in the same sense than any Boolean algebra is a direct product of two elements Boolean algebras). This representation theorem implies that, to prove that an equation holds in *all* MV–algebras it is enough to show that it holds in $\mathcal{L}$. This fact is used in proving the following three propositions.

Proposition 1. *In an MV–algebra* $\mathbf{L}$ *the following holds for all* $x, y \in L$

$$(x \odot y) \wedge (x^* \odot y^*) = \mathbf{0}, \tag{30}$$

$$(x^* \wedge y) \oplus (x \odot y) \oplus (x^* \odot y^*) \oplus (x \wedge y^*) = \mathbf{1}. \tag{31}$$

Proposition 2. *Assume* x, y, a, b *are elements of an MV–algebra* $\mathbf{L}$ *such that the following system of equations holds*

$$\text{(A)} \begin{cases} x^* \wedge y &= a^* \wedge b, \\ x \odot y &= a \odot b, \\ x^* \odot y^* &= a^* \odot b^*, \\ x \wedge y^* &= a \wedge b^*. \end{cases}$$

Then $x = a$ *and* $y = b$.

Proposition 3. *Assume* x, y *are elements of an MV–algebra* $\mathbf{L}$ *such that*

$$\text{(B)} \begin{cases} x^* \wedge y &= f, \\ x \odot y &= k, \\ x^* \odot y^* &= u, \\ x \wedge y^* &= t. \end{cases}$$

Then (C) $x = t \oplus k$, $y = f \oplus k$ *and (D)* $x = (f \oplus u)^*$, $y = (t \oplus u)^*$.

Propositions 2 and 3 put ordered couples $\langle x, y\rangle$ and values f, k, u, t defined by (B) into a one–one correspondence.

Definition 1. *A complete MV-algebra* $\mathbf{L}$ *is* injective (cf. [6]) *if, for any* $a \in L$ *and any natural number* n*, there is an element* $b \in L$*, called the* n–divisor of a*, such that* $nb = \underbrace{b \oplus \cdots \oplus b}_{n \text{ times}} = a$ *and* $(a^* \oplus (n-1)b)^* = b$.

All n–divisors are unique (cf. [8]). The Lukasiewicz structure $\mathcal{L}$ and all finite Lukasiewicz chains are injective MV–algebras (cf. [13]).

3 Evidence Couples and Evidence Matrices

Let $\mathbf{L} = \langle L, \oplus, ^*, \mathbf{0}\rangle$ be an MV–algebra. The product set $L \times L$ can be equipped with an MV–structure by setting

$$\langle a, b\rangle \otimes \langle c, d\rangle = \langle a \oplus c, b \odot d\rangle, \tag{32}$$

$$\langle a, b\rangle^{\perp} = \langle a^*, b^*\rangle, \tag{33}$$

$$\overline{\mathbf{0}} = \langle \mathbf{0}, \mathbf{1}\rangle \tag{34}$$

for each ordered couple $\langle a, b\rangle, \langle c, d\rangle \in L \times L$. Indeed, the axioms (5)–(9) hold trivially and, to prove that the axiom (10) holds, it is enough to realize that

$$\begin{aligned}(\langle a, b\rangle^{\perp} \otimes \langle c, d\rangle)^{\perp} \otimes \langle c, d\rangle &= \langle a \vee c, b \wedge d\rangle = \langle c \vee a, d \wedge b\rangle \\ &= (\langle c, d\rangle^{\perp} \otimes \langle a, b\rangle)^{\perp} \otimes \langle a, b\rangle.\end{aligned}$$

It is routine to verify that the order on $L \times L$ is defined via

$$\langle a, b\rangle \leq \langle c, d\rangle \text{ if and only if } a \leq x, y \leq b, \tag{35}$$

the lattice operation by

$$\langle a, b\rangle \vee \langle c, d\rangle = \langle a \vee c, b \wedge d\rangle, \tag{36}$$

$$\langle a, b\rangle \wedge \langle c, d\rangle = \langle a \wedge c, b \vee d\rangle, \tag{37}$$

and an adjoin couple $\langle \star, \mapsto\rangle$ by

$$\langle a, b\rangle \star \langle c, d\rangle = \langle a \odot c, b \oplus d\rangle, \tag{38}$$

$$\langle a, b\rangle \mapsto \langle c, d\rangle = \langle a \rightarrow c, (d \rightarrow b)^*\rangle. \tag{39}$$

Notice that $a \rightarrow c = a^* \oplus c$ and $(d \rightarrow b)^* = (d^* \oplus b)^* = d \odot b^* = b^* \odot d$.

Definition 2. *Given an MV-algebra* $\mathbf{L}$*, denote the structure described via (32) - (39) by* $\mathbf{L}_{EC}$ *and call it the* MV–algebra of evidence couples induced by $\mathbf{L}$.

Definition 3. *Given an MV-algebra* $\mathbf{L}$*, denote*

$$\mathcal{M} = \left\{ \begin{bmatrix} a^* \wedge b & a \odot b \\ a^* \odot b^* & a \wedge b^* \end{bmatrix} | \langle a, b\rangle \in L \times L \right\}$$

and call it the set of evidence matrices induced by evidence couples.

By Proposition 2 we have

Theorem 1. *There is a one–to–one correspondence between* $L \times L$ *and* $\mathcal{M}$*: if* $A, B \in \mathcal{M}$ *are two evidence matrices induced by evidence couples* $\langle a, b\rangle$ *and* $\langle x, y\rangle$*, respectively, then* $A = B$ *if and only if* $a = x$ *and* $b = y$.

The MV–structure descends from $\mathbf{L}_{EC}$ to $\mathcal{M}$ in a natural way: if $A, B \in \mathcal{M}$ are two evidence matrices induced by evidence couples $\langle a, b\rangle$ and $\langle x, y\rangle$, respectively, then the evidence couple $\langle a \oplus x, b \odot y\rangle$ induces an evidence matrix

$$C = \begin{bmatrix} (a \oplus x)^* \wedge (b \odot y) & (a \oplus x) \odot (b \odot y) \\ (a \oplus x)^* \odot (b \odot y)^* & (a \oplus x) \wedge (b \odot y)^* \end{bmatrix}.$$

Thus, we may define a binary operation $\bigoplus$ on $\mathcal{M}$ by

$$\begin{bmatrix} a^* \wedge b & a \odot b \\ a^* \odot b^* & a \wedge b^* \end{bmatrix} \bigoplus \begin{bmatrix} x^* \wedge y & x \odot y \\ x^* \odot y^* & x \wedge y^* \end{bmatrix} = C.$$

Similarly, if $A \in \mathcal{M}$ is an evidence matrix induced by an evidence couple $\langle a, b\rangle$, then the evidence couple $\langle a^*, b^*\rangle$ induces an evidence matrix

$$A^{\perp} = \begin{bmatrix} a \wedge b^* & a^* \odot b^* \\ a \odot b & a^* \wedge b \end{bmatrix}.$$

In particular, the evidence couple $\langle \mathbf{0}, \mathbf{1}\rangle$ induces the following evidence matrix

$$F = \begin{bmatrix} \mathbf{0}^* \wedge \mathbf{1} & \mathbf{0} \odot \mathbf{1} \\ \mathbf{0}^* \odot \mathbf{1}^* & \mathbf{0} \wedge \mathbf{1}^* \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Theorem 2. *Let* $\mathbf{L}$ *be an MV–algebra. The structure* $\mathcal{M} = \langle \mathcal{M}, \bigoplus, ^{\perp}, F\rangle$ *as defined above is an MV-algebra (called* the MV–algebra of evidence matrices*).*

$$\text{Assume } A = \begin{bmatrix} a^* \wedge b & a \odot b \\ a^* \odot b^* & a \wedge b^* \end{bmatrix}, \ B = \begin{bmatrix} x^* \wedge y & x \odot y \\ x^* \odot y^* & x \wedge y^* \end{bmatrix} \in \mathcal{M}$$

Then it is obvious that the lattice operations $\wedge$, $\vee$, the monoidal operation $\bigodot$ and the residual operation $\longrightarrow$ are defined via

$$A \wedge B = \begin{bmatrix} (a \wedge x)^* \wedge (b \vee y) & (a \wedge x) \odot (b \vee y) \\ (a \wedge x)^* \odot (b \vee y)^* & (a \wedge x) \wedge (b \vee y)^* \end{bmatrix},$$

$$A \vee B = \begin{bmatrix} (a \vee x)^* \wedge (b \wedge y) & (a \vee x) \odot (b \wedge y) \\ (a \vee x)^* \odot (b \wedge y)^* & (a \vee x) \wedge (b \wedge y)^* \end{bmatrix},$$

$$A \bigodot B = \begin{bmatrix} (a \odot x)^* \wedge (b \oplus y) & (a \odot x) \odot (b \oplus y) \\ (a \odot x)^* \odot (b \oplus y)^* & (a \odot x) \wedge (b \oplus y)^* \end{bmatrix},$$

$$A \longrightarrow B = \begin{bmatrix} (a \to x)^* \wedge (y \to b)^* & (a \to x) \odot (y \to b)^* \\ (a \to x)^* \odot (y \to b) & (a \to x) \wedge (y \to b) \end{bmatrix}.$$

If the original MV–algebra $\mathbf{L}$ is complete, then the structure $\mathcal{M}$ is a complete MV–algebra, too, and supremes and infimas are defined by evidence couples

$$\bigvee_{i\in\Gamma}\{\langle a_i, b_i\rangle\} = \langle \bigvee_{i\in\Gamma} a_i, \bigwedge_{i\in\Gamma} b_i\rangle\}, \quad \bigwedge_{i\in\Gamma}\{\langle a_i, b_i\rangle\} = \langle \bigwedge_{i\in\Gamma} a_i, \bigvee_{i\in\Gamma} b_i\rangle\}.$$

Thus, we may define another residual operation $\Rightarrow$ on $\mathcal{M}$ via

$$A \Rightarrow B = \begin{bmatrix} (a \Rightarrow x)^* \wedge (b^* \Rightarrow y^*)^* & (a \Rightarrow x) \odot (b^* \Rightarrow y^*)^* \\ (a \Rightarrow x)^* \odot (b^* \Rightarrow y^*) & (a \Rightarrow x) \wedge (b^* \Rightarrow y^*) \end{bmatrix}.$$

To verify this last claim, assume $\langle a, b\rangle \wedge \langle x, y\rangle \leq \langle c, d\rangle$ in $\mathbf{L}_{EC}$, which is equivalent to

$$a \wedge x \leq c \text{ and } d \leq b \vee y, \text{ that is,}$$

$$a \leq x \Rightarrow c \text{ and } (b \vee y)^* = b^* \wedge y^* \leq d^*, \text{ i.e.,}$$

$$a \leq x \Rightarrow c \text{ and } b^* \leq y^* \Rightarrow d^*, \text{ or equivalently,}$$

$$a \leq x \Rightarrow c \text{ and } (y^* \Rightarrow d^*)^* \leq b, \text{ i.e.,}$$

$\langle a, b\rangle \leq \langle x \Rightarrow c, (y^* \Rightarrow d^*)^*\rangle$ in $\mathbf{L}_{EC}$. Therefore, if A is induced by $\langle a, b\rangle$ and B is induced by $\langle x, y\rangle$ then the evidence matrix $A \Rightarrow B$ is induced by the evidence couple $\langle a \Rightarrow x, (b^* \Rightarrow y^*)^*\rangle$. In particular, the weak complementation * on $\mathcal{M}$ is defined via $A^\star = A \Rightarrow F$ and induced by

$$\begin{array}{l}
\langle \mathbf{1}, \mathbf{0}\rangle \text{ if } a = \mathbf{0},\ b = \mathbf{1}, \text{ then } A^\star = T,\\
\langle \mathbf{0}, \mathbf{0}\rangle \text{ if } a > \mathbf{0},\ b = \mathbf{1}, \text{ then } A^\star = U,\\
\langle \mathbf{1}, \mathbf{1}\rangle \text{ if } a = \mathbf{0},\ b < \mathbf{1}, \text{ then } A^\star = K,\\
\langle \mathbf{0}, \mathbf{1}\rangle \text{ if } a > \mathbf{0},\ b < \mathbf{1}, \text{ then } A^\star = F.
\end{array}$$

The matrices F, T, K, U correspond to Belnap's original values *false, true, contradictory, unknown*, respectively.

Theorem 3. $\mathbf{L}$ *is an injective MV–algebra if, and only if the corresponding MV–algebra of evidence matrices* $\mathcal{M}$ *is an injective MV–algebra.*

4 Para Consistent Pavelka Style Fuzzy Logic

4.1 Pavelka Style Fuzzy Logic

A standard approach in mathematical sentential logic is the following. After introducing atomic formulae, logical connectives and the set of well–formed formulae, these formulae are semantically interpreted in suitable algebraic structures. In Classical logic these structures are Boolean algebras, in Hájek's Basic fuzzy logic [7], for example, the suitable structures are BL–algebras. *Tautologies* of a logic are those formulae that obtain the top value $\mathbf{1}$ in all interpretations in all suitable algebraic structures; for this reason tautologies are sometimes called $\mathbf{1}$-tautologies. For example, tautologies in Basic fuzzy logic are exactly the formulae that obtain value $\mathbf{1}$ in all interpretations in all BL–algebras. The standard next step in mathematical sentential logic is to fix the axiom scheme and the rules of inference: a well–formed formula is a *theorem* if it is either an axiom or obtained recursively from axioms by using finite many times rules of inference. *Completeness* of the logic means that tautologies and theorems coincide; Classical sentential logic and Basic fuzzy sentential logic, for example, are complete logics.

In Pavelka style fuzzy sentential logic [10] the situation is somewhat different. We start by fixing a set of truth values, in fact an algebraic structure – in Pavelka's own approach this structure in the Lukasiewicz structure $\mathcal{L}$ on the real unit interval while in [13] the structure is a more general (but fixed!) injective MV–algebra $\mathbf{L}$. In this brief review we follow [13].

Consider a zero order language $\mathcal{F}$ with a set of infinite many propositional variables $\mathtt{p}, \mathtt{q}, \mathtt{r}, \cdots$, and a set of *inner truth values* $\{\mathbf{a} \mid a \in L\}$ corresponding to elements in the set L. Proved in [7], if the set of truth values is the whole

real interval $[0, 1]$ then it is enough to include inner truth values corresponding to rationals $\in [0, 1]$. In two–valued logic inner truth values correspond to the truth constants $\perp$ and $\top$. These two sets of objects constitute the set $\mathcal{F}_a$ of *atomic formulae*. The elementary logical connectives are *implication* '`imp`' and *conjunction* '`and`'. The set of all well formed formulae (wffs) is obtained in the natural way: atomic formulae are wffs and if α, β are wffs, then 'α `imp` β', 'α `and` β' are wffs.

As shown in [13], we can introduce many other logical connectives by abbreviations, e.g. *disjunction* '`or`', *negation* '`non`', *equivalence* '`equiv`' and *exclusive or* '`xor`' are possible. Also the connectives *weak implication* '$\overline{\texttt{imp}}$', *weak conjunction* '$\overline{\texttt{and}}$', *weak disjunction* '$\overline{\texttt{or}}$', *weak negation* '$\overline{\texttt{non}}$', *weak equivalence* '$\overline{\texttt{equiv}}$' and *weak exclusive or* '$\overline{\texttt{xor}}$' are available in this logic. We call the logical connectives without bar *Lukasiewicz* connectives, those with bar are *Intuitionistic* connectives.

Semantics in Pavelka style fuzzy sentential logic is introduced in the following way: any mapping $v : \mathcal{F}_a \mapsto L$ such that $v(\mathbf{a}) = a$ for all inner truth values $\mathbf{a}$ can be extended recursively into the whole $\mathcal{F}$ by setting $v(\alpha \texttt{ imp } \beta) = v(\alpha) \to v(\beta)$ and $v(\alpha \texttt{ and } \beta) = v(\alpha) \odot v(\beta)$. Such mappings v are called *valuations*. The *degree of tautology* of a wff α is the infimum of all values $v(\alpha)$, that is

$$\mathcal{C}^{sem}(\alpha) = \bigwedge\{v(\alpha)|\ v \text{ is a valuation }\}.$$

We may also fix some set $T \subseteq \mathcal{F}$ of wffs and consider valuations v such that $T(\alpha) \leq v(\alpha)$ for all wffs α. If such a valuation exists, the T is called *satisfiable*. We say that T is a *fuzzy theory* and formulae α such that $T(\alpha) \neq 0$ are the *non–logical axioms* of the fuzzy theory T. Then we consider values

$$\mathcal{C}^{sem}(T)(\alpha) = \bigwedge\{v(\alpha)|\ v \text{ is a valuation, } T \text{ satisfies } v\}.$$

The set of logical axioms, denoted by `A`, is composed by the eleven forms of formulae listed on page 88 in [13]. A *fuzzy rule of inference* is a scheme

$$\frac{\alpha_1, \cdots, \alpha_n}{r^{\mathtt{syn}}(\alpha_1, \cdots, \alpha_n)}, \frac{a_1, \cdots, a_n}{r^{\mathtt{sem}}(\alpha_1, \cdots, \alpha_n)},$$

where the wffs $\alpha_1, \cdots, \alpha_n$ are *premises* and the wff $r^{\mathtt{syn}}(\alpha_1, \cdots, \alpha_n)$ is the *conclusion*. The values $a_1, \cdots, a_n$ and $r^{\mathtt{sem}}(\alpha_1, \cdots, \alpha_n) \in L$ are the corresponding truth values. The mappings $L^n \mapsto L$ are semi–continuous, i.e.

$$r^{\mathtt{sem}}(\alpha_1, \cdots, \textstyle\bigvee_{j\in\Gamma} a_{k_j}, \cdots, \alpha_n) = \textstyle\bigvee_{j\in\Gamma} r^{\mathtt{sem}}(\alpha_1, \cdots, a_{k_j}, \cdots, \alpha_n)$$

holds for all $1 \leq k \leq n$. Moreover, the fuzzy rules are required to be *sound* in a sense that

$$r^{\mathtt{sem}}(v(\alpha_1), \cdots, v(\alpha_n)) \leq v(r^{\mathtt{syn}}(\alpha_1, \cdots, \alpha_n))$$

holds for all valuations v.

The following are examples of fuzzy rules of inference, denoted by a set `R`:

Generalized Modus Ponens:

$$\frac{\alpha, \alpha \texttt{ imp } \beta}{\beta}, \frac{a, b}{a \odot b}$$

a–Consistency testing rules :

$$\frac{\mathbf{a}}{\mathbf{0}}\,,\,\frac{b}{c}$$

where **a** is an inner truth value and $c = \mathbf{0}$ if $b \le a$ and $c = \mathbf{1}$ elsewhere.

a–Lifting rules :

$$\frac{\alpha}{\mathbf{a}\ \texttt{imp}\ \alpha}\,,\,\frac{b}{a \to b}$$

where **a** is an inner truth value.

Rule of Bold Conjunction:

$$\frac{\alpha, \beta}{\alpha\ \texttt{and}\ \beta}\,,\,\frac{A, B}{A \odot B}$$

A *meta proof* w of a wff α in a fuzzy theory T is a finite sequence

$$\begin{array}{ccc} \alpha_1 & , & a_1 \\ \vdots & & \vdots \\ \alpha_m & , & a_m \end{array}$$

where
(i) $\alpha_m = \alpha$,
(ii) for each i, $1 \le i \le m$, α_i is a logical axiom, or is a non–logical axiom, or there is a fuzzy rule of inference in $\mathtt{R}$ and wff formulae $\alpha_{i_1}, \cdots, \alpha_{i_n}$ with $i_1, \cdots, i_n < i$ such that $\alpha_i = r^{\mathtt{syn}}(\alpha_{i_1}, \cdots, \alpha_{i_n})$,
(iii) for each i, $1 \le i \le m$, the value $a_i \in L$ is given by

$$a_i = \begin{cases} a & \text{if } \alpha_i \text{ is the axiom } \mathbf{a} \\ 1 & \text{if } \alpha_i \text{ is some other logical axiom in the set } \mathtt{A} \\ T(\alpha_i) & \text{if } \alpha_i \text{ is a non–logical axiom} \\ r^{\mathtt{sem}}(a_{i_1}, \cdots, a_{i_n}) & \text{if } \alpha_i = r^{\mathtt{syn}}(\alpha_{i_1}, \cdots, \alpha_{i_n}) \end{cases}$$

The value a_m is called the *degree* of the meta proof w. Since a wff α may have various meta proofs with different degrees, we define the *degree of deduction* of a formula α to be the supremum of all such values, i.e.,

$$\mathcal{C}^{syn}(T)(\alpha) = \bigvee\{a_m |\ w \text{ is a meta proof for } \alpha \text{ in the fuzzy theory } T\}.$$

A fuzzy theory T is *consistent* if $\mathcal{C}^{sem}(T)(\mathbf{a}) = a$ for all inner truth values **a**. By Proposition 94 in [13], any satisfiable fuzzy theory is consistent. Theorem 25 in [13] now states the completeness of Pavelka style sentential logic:

If a fuzzy theory T is consistent, then $\mathcal{C}^{sem}(T)(\alpha) = \mathcal{C}^{syn}(T)(\alpha)$ for any wff α.

Thus, in Pavelka style fuzzy sentential logic we may talk about tautologies of a degree a and theorems of a degree a for all truth values $a \in L$, and these concepts coincide. This completeness result remains valid if we extend the language to contain Intuitionistic connectives '$\overline{\texttt{and}}$' or '$\overline{\texttt{or}}$'. However, it does not hold if the language is extended by the Intuitionistic connectives '$\overline{\texttt{imp}}$' or '$\overline{\texttt{non}}$'.

4.2 Para Consistent Pavelka Logic

The above construction can be carried out in any injective MV–algebra thus, in particular, in the injective MV–algebra $\mathcal{M}$ of evidence matrices induced by an injective MV–algebra $\mathbf{L}$. Indeed, semantics is introduced by associating to each atomic formula $\mathtt{p}$ an evidence couple $\langle \mathtt{pro}, \mathtt{con} \rangle$ or simply $\langle a, b \rangle \in \mathbf{L}_{EC}$. The evidence couple $\langle a, b \rangle$ induces a unique evidence matrix $A \in \mathcal{M}$ and therefore *valuations* are mappings v such that $v(\mathtt{p}) = A$ for propositional variables and $v(\mathbf{I}) = I$ for inner truth values ($\in \mathcal{M}$). A valuation v is then extended recursively to whole $\mathcal{F}$ via

$$v(\alpha \ \mathtt{imp}\ \beta) = v(\alpha) \longrightarrow v(\beta), \qquad v(\alpha \ \mathtt{and}\ \beta) = v(\alpha) \bigodot v(\beta). \tag{40}$$

Similar to the procedure in [13], Chapter 3.1, we can show that

$$v(\alpha \ \mathtt{or}\ \beta) = v(\alpha) \bigoplus v(\beta), \qquad v(\mathtt{non}{-}\alpha) = [v(\alpha)]^{\perp}, \tag{41}$$

$$v(\alpha \ \mathtt{equiv}\ \beta) = [v(\alpha) \longrightarrow v(\beta)] \wedge [v(\beta) \longrightarrow v(\alpha)], \tag{42}$$

$$v(\alpha \ \mathtt{xor}\ \beta) = [v(\alpha) \bigoplus v(\beta)] \wedge [v(\beta) \longrightarrow v(\alpha)^{\perp}] \wedge [v(\alpha) \longrightarrow v(\beta)^{\perp}], \tag{43}$$

$$v(\alpha \ \overline{\mathtt{and}}\ \beta) = v(\alpha) \wedge v(\beta), \quad v(\alpha \ \overline{\mathtt{or}}\ \beta) = v(\alpha) \vee v(\beta), \tag{44}$$

$$v(\alpha \ \overline{\mathtt{imp}}\ \beta) = v(\alpha) \Rightarrow v(\beta), \qquad v(\overline{\mathtt{non}{-}}\alpha) = v(\alpha)^{\star}, \tag{45}$$

$$v(\alpha \ \overline{\mathtt{equiv}}\ \beta) = [v(\alpha) \Rightarrow v(\beta)] \wedge [v(\beta) \Rightarrow v(\alpha)]. \tag{46}$$

The obtained continuous valued para consistent logic is a complete logic in the Pavelka sense. The logical axioms and the rules of inference are those defined in [13], Chapter 3. Thus, we have a solid syntax available and e.g. all the many–valued extensions of classical rules of inference are available; 25 such rules are listed in [13].

If the MV–algebra $\mathbf{L}$ is the Lukasiewicz structure, then the evidence couples coincide with the ordered pairs $\langle B(\alpha), B(\neg\alpha) \rangle$ discussed in [11]. Moreover, the evidence matrices coincide with the matrices

$$v(\alpha) = \begin{bmatrix} f(\alpha) & k(\alpha) \\ u(\alpha) & t(\alpha) \end{bmatrix},$$

where $t(\alpha)$, $k(\alpha)$, $u(\alpha)$, $f(\alpha)$ are defined via equations (1) – (4) (equations (38) – (41) in [11]). In particular, the computation of values $v(\alpha \wedge \beta)$ and $v(\alpha \vee \beta)$ (Proposition 3.3. in [11]) coincide with our equations (44)).

References

[1] Arieli, O., Avron, A.: Reasoning with logical bilattices. Journal of Logic, Language and Information 5, 25–63 (1996)

[2] Belnap, N.D.: A useful four–valued logic. In: Epstein, G., Dumme, J. (eds.) Modern uses of multiple valued logics, pp. 8–37. D. Reidel, Dordrecht (1977)

[3] Chang, C.C.: Algebraic analysis of many-valued logics. Transactions of the American Mathematical Society 88, 476–490 (1958)

[4] Dubois, D.: On Ignorance and Contradiction Considered as Truth–values. Logic Journal of IGPL 16(2), 195–216 (2008)
[5] Ginsberg, M.K.: Multivalued Logics: A Uniform Approach to Inference in Artificial Intelligence. Computational Intelligence 4, 265–316 (1988)
[6] Gluschankof, D.: Prime deductive systems and injective objects in the algebras of Lukasiewicz infinite–valued calculi. Algebra Universalis 29, 354–377 (1992)
[7] Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
[8] Kukkurainen, P., Turunen, E.: Many-valued Similarity Reasoning. An Axiomatic Approach International Journal of Multiple Valued Logic 8, 751–760 (2002)
[9] Cignoli, R., D'Ottaviano, I.M.L., Mundici, D.: Algebraic Foundations of many-valued Reasoning. In: Trends in Logic, vol. 7. Kluwer Academic Publishers, Dordrecht (2000)
[10] Pavelka, J.: On fuzzy logic I, II, III. Zeitsch. f. Math. Logik 25, 45–52, 119–134, 447–464 (1979)
[11] Perny, P., Tsoukias, A.: On the continuous extensions of four valued logic for preference modeling. In: Proceedings of the IPMU conference, pp. 302–309 (1998)
[12] Tsoukias, A.: A first order, four valued, weakly Para consistent logic and its relation to rough sets semantics. Foundations of Computing and Decision Sciences 12, 85–108 (2002)
[13] Turunen, E.: Mathematics behind Fuzzy Logic. Springer, Heidelberg (1999)
[14] Öztürk, M., Tsoukias, A.: Modeling uncertain positive and negative reasons in decision aiding. Decision Support Systems 43, 1512–1526 (2007)

From Philosophical to Industrial Logics*

Moshe Y. Vardi**

Rice University, Department of Computer Science, Rice University,
Houston, TX 77251-1892, USA
vardi@cs.rice.edu
http://www.cs.rice.edu/~vardi

Abstract. One of the surprising developments in the area of program verification is how ideas introduced by logicians in the early part of the 20th Century ended up yielding by the 21 Century industrial-standard property-specification languages. This development was enabled by the equally unlikely transformation of the mathematical machinery of automata on infinite words, introduced in the early 1960s for second-order logic, into effective algorithms for model-checking tools. This paper attempts to trace the tangled threads of this development.

1 Thread I: Classical Logic of Time

1.1 Monadic Logic

In 1902, Russell send a letter to Frege in which he pointed out that Frege's logical system was inconsistent. This inconsistency has become known as *Russell's Paradox*. Russell, together with Whitehead, published *Principia Mathematica* in an attempt to resolve the inconsistency, but the monumental effort did not convince mathematicians that mathematics is indeed free of contradictions. This has become know as the "Foundational Crisis." In response to that Hilbert launched what has become known as "Hilbert's Program." (See [1].)

One of the main points in Hilbert's program was the decidability of mathematic. In 1928, Hilbert and Ackermann published "Principles of Mathematical Logic", in which they posed the question of the *Decision Problem* for first-order logic. This problem was shown to be unsolvable by Church and Turing, independently, in 1936; see [2]. In response to that, logicians started the project of classifying the decidable fragments of first-order logic [2,3]. The earliest decidability result for such a fragment is for the *Monadic Class*, which is the fragment of first-order predicate logic where all predicates, with the exception of the equality predicate, are required to be monadic. This fragment can express the classical sylogisms. For example the formula

$$((\forall x)(H(x) \rightarrow M(x)) \wedge (\forall x)(G(x) \rightarrow H(x))) \rightarrow (\forall x)(G(x) \rightarrow M(x))$$

* A earlier version of this paper, under the title "From Church and Prior to PSL", appeared in the *Proc. 2006 Workshop on 25 Years of Model Checking, Lecture Notes in Computer Science, Springer.*

** Supported in part by NSF grants CCR-9988322, CCR-0124077, CCR-0311326, and ANI-0216467, by BSF grant 9800096, and by a gift from the Intel Corporation.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 89–115, 2009.

expresses the inference of: "if all humans are mortal and all Greeks are human, then all Greeks are mortal."

In 1915 Löwenheim showed that the Monadic Class is decidable [4]. His proof technique was based on the *bounded-model property*, proving that a monadic sentence is satisfiable if it is satisfiable in a model of bounded size. This enables the reduction of satisfiability testing to searching for a model of bounded size. L"owenheim's tecchnique was extended by Skolem in 1919 to *Monadic Second Order Logic*, in which one can also quantify over monadic predicates, in addition to quantifying over domain elements [5]. Skolem also used the bounded-model property. To prove this property, he introduced the technique of *quantifier elimination*, which is a key technique in mathematical logic [2].

Recall, that the only binary predicate in Skolem's monadic second-order logic is the equality predicate. One may wonder what happens if we also allow inequality predicates. Such an extension is the subject of the next section.

1.2 Logic and Automata

Classical logic views logic as a declarative formalism, aimed at the specification of properties of mathematical objects. For example, the sentence

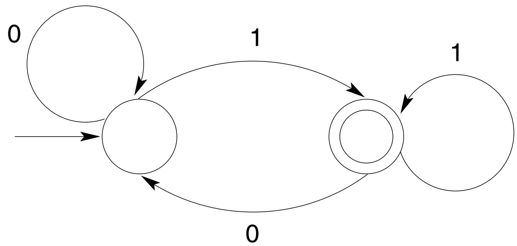
$$(\forall x, y, x)(mult(x, y, z) \leftrightarrow mult(y, x, z))$$

expressed the commutativity of multiplication. Starting in the 1930s, a different branch of logic focused on formalisms for describing computations, starting with the introduction of Turing machines in the 1930s, and continuing with the development of the theory of finite-state machines in the 1950s. A surprising, intimate, connection between these two paradigms of logic emerged in the late 1950s.

A *nondeterministic finite automaton on words* (NFW) $A = (\Sigma, S, S_0, \rho, F)$ consists of a finite input alphabet Σ, a finite state set S, an initial state set $S_0 \subseteq S$, a transition relation $\rho \subseteq S \times \Sigma \times S$, and an accepting state set $F \subseteq S$. An NFW runs over an finite input word $w = a_0, \ldots, a_{n-1} \in \Sigma^*$. A *run* of A on w is a finite sequence $r = s_0, \ldots, s_n$ of states in S such that $s_0 \in S_0$, and $(s_i, a_i, s_{i+1}) \in \rho$, for $0 \leq i < n$. The run r is *accepting* if $s_n \in F$. The word w is *accepted* by A if A has an accepting run on w. The *language* of A, denoted $L(A)$, is the set of words accepted by A. The class of languages accepted by NFWs forms the class of *regular* languages, which are defined in terms of regular expressions. This class is extremely robust and has numerous equivalent representations [6].

Example 1. We describe graphically below an NFW that accepts all words over the alphabet $\{0, 1\}$ that end with an occurrence of 1. The arrow on the left designates the initial state, and the circle on the right designates an accepting state.

We now view a finite word $w = a_0, \ldots, a_{n-1}$ over an alphabet Σ as a relational structure M_w, with the domain of $0, \ldots, n-1$ ordered by the binary relation $<$, and the unary relations $\{P_a : a \in \Sigma\}$, with the interpretation that $P_a(i)$ holds



precisely when $a_i = a$. We refer to such structures as *word structures*. We now use first-order logic (FO) to talk about such words. For example, the sentence

$$(\exists x)((\forall y)(\neg(x < y)) \wedge P_a(x))$$

says that the last letter of the word is a. We say that such a sentence is over the alphabet Σ.

Going beyond FO, we obtain *monadic second-order logic* (MSO), in which we can have monadic second-order quantifiers of the form $\exists Q$, ranging over subsets of the domain, and giving rise to new atomic formulas of the form $Q(x)$. Given a sentence φ in MSO, its set of models models(φ) is a set of words. Note that this logic extends Skolem's logic with the addition of the linear order $<$.

The fundamental connection between logic and automata is now given by the following theorem, discovered independently by Büchi, Elgot, and Trakhtenbrot.

Theorem 1. [7,8,9,10,11,12] *Given an MSO sentence φ over alphabet Σ, one can construct an NFW A_φ with alphabet Σ such that a word w in Σ^* is accepted by A_φ iff φ holds in the word structure M_w. Conversely, given an NFW A with alphabet Σ, one can construct an MSO sentence φ_A over Σ such that φ_A holds in a word structure M_w iff w is accepted by A.*

Thus, the class of languages defined by MSO sentences is precisely the class of regular languages.

To decide whether a sentence φ is *satisfiable*, that is, whether models(φ) $\neq \emptyset$, we need to check that $L(A_\varphi) \neq \emptyset$. This turns out to be an easy problem. Let $A = (\Sigma, S, S_0, \rho, F)$ be an NFW. Construct a directed graph $G_A = (S, E_A)$, with S as the set of nodes, and $E_A = \{(s,t) : (s,a,t) \in \rho$ for some $a \in \Sigma\}$. The following lemma is implicit in [7,8,9,10] and more explicit in [13].

Lemma 1. *$L(A) \neq \emptyset$ iff there are states $s_0 \in S_0$ and $t \in F$ such that in G_A there is a path from s_0 to t.*

We thus obtain an algorithm for the Satisfiability problem of MSO over word structures: given an MSO sentence φ, construct the NFW A_φ and check whether $L(A) \neq \emptyset$ by finding a path from an initial state to an accepting state. This approach to satisfiability checking is referred to as the *automata-theoretic approach*,

since the decision procedure proceeds by first going from logic to automata, and then searching for a path in the constructed automaton.

There was little interest in the 1950s in analyzing the computational complexity of the SATISFIABILITY problem. That had to wait until 1974. Define the function $exp(k, n)$ inductively as follows: $exp(0, n) = n$ and $exp(k+1, n) = 2^{exp(k,n)}$. We say that a problem is *nonelementary* if it can not be solved by an algorithm whose running time is bounded by $exp(k, n)$ for some fixed $k \geq 0$; that is, the running time cannot be bounded by a tower of exponentials of a fixed height. It is not too difficult to observe that the construction of the automaton A_φ in [7,8,9,10] involves a blow-up of $exp(n, n)$, where n is the length of the MSO sentence being decided. It was shown in [14,15] that the SATISFIABILITY problem for MSO is nonelementary. In fact, the problem is already nonelementary for FO [15].

1.3 Reasoning about Sequential Circuits

The field of hardware verification seems to have been started in a little known 1957 paper by Church, in which he described the use of logic to specify *sequential circuits* [16]. A sequential circuit is a switching circuit whose output depends not only upon its input, but also on what its input has been in the past. A sequential circuit is a particular type of finite-state machine, which became a subject of study in mathematical logic and computer science in the 1950s.

Formally, a sequential circuit $C = (I, O, R, f, g, \mathbf{r}_0)$ consists of a finite set I of Boolean input signals, a finite set O of Boolean output signals, a finite set R of Boolean sequential elements, a transition function $f : 2^I \times 2^R \to 2^R$, an output function $g : 2^R \to 2^O$, and an initial state $\mathbf{r}_0 \in 2^R$. (We refer to elements of $I \cup O \cup R$ as *circuit elements*, and assume that I, O, and R are disjoint.) Intuitively, a state of the circuit is a Boolean assignment to the sequential elements. The initial state is $\mathbf{r}_0$. In a state $\mathbf{r} \in 2^R$, the Boolean assignment to the output signals is $g(\mathbf{r})$. When the circuit is in state $\mathbf{r} \in 2^R$ and it reads an input assignment $\mathbf{i} \in 2^I$, it changes its state to $f(\mathbf{i}, \mathbf{r})$.

A *trace* over a set V of Boolean variables is an infinite word over the alphabet 2^V, i.e., an element of $(2^V)^\omega$. A trace of the sequential circuit C is a trace over $I \cup O \cup R$ that satisfies some conditions. Specifically, a sequence $\tau = (\mathbf{i}_0, \mathbf{r}_0, \mathbf{o}_0)$, $(\mathbf{i}_1, \mathbf{r}_1, \mathbf{o}_1), \ldots$, where $\mathbf{i}_j \in 2^I$, $\mathbf{o}_j \in 2^O$, and $\mathbf{r}_j \in 2^R$, is a trace of C if $\mathbf{r}_{j+1} = f(\mathbf{i}_j, \mathbf{r}_j)$ and $\mathbf{o}_j = g(\mathbf{r}_j)$, for $j \geq 0$. Thus, in modern terminology, Church was following the *linear-time* approach [17] (see discussion in Section 2.1). The set of traces of C is denoted by traces(C).

We saw earlier how to associate relational structures with words. We can similarly associate with an infinite word $w = a_0, a_1, \ldots$ over an alphabet 2^V, a relational structure $M_w = (\mathrm{N}, \leq, V)$, with the naturals N as the domain, ordered by $<$, and extended by the set V of unary predicates, where $j \in p$, for $p \in V$, precisely when p *holds* (i.e., is assigned 1) in a_i.[1] We refer to such structures as *infinite word structures*. When we refer to the *vocabulary* of such a structure, we refer explicitly only to V, taking $<$ for granted.

[1] We overload notation here and treat p as both a Boolean variable and a predicate.

We can now specify traces using First-Order Logic (FO) sentences constructed from atomic formulas of the form $x = y$, $x < y$, and $p(x)$ for $p \in V = I \cup R \cup O$.[2] For example, the FO sentence

$$(\forall x)(\exists y)(x < y \wedge p(y))$$

says that p holds infinitely often in the trace. In a follow-up paper in 1963 [18], Church considered also specifying traces using monadic second-order logic (MSO), where in addition to first-order quantifiers, which range over the elements of $\mathbb{N}$, we allow also monadic second-order quantifiers, ranging over subsets of $\mathbb{N}$, and atomic formulas of the form $Q(x)$, where Q is a monadic predicate variable. (This logic is also called *S1S*, the "second-order theory of one successor function".) For example, the MSO sentence,

$$\begin{aligned}(\exists P)(\forall x)(\forall y)((((P(x) \wedge y = x+1) \rightarrow (\neg P(y)))\wedge \\ (((\neg P(x)) \wedge y = x+1) \rightarrow P(y)))\wedge \\ (x = 0 \rightarrow P(x)) \wedge (P(x) \rightarrow q(x))),\end{aligned}$$

where $x = 0$ is an abbrevaition for $(\neg(\exists z)(z < x))$ and $y = x+1$ is an abbreviation for $(y > x \wedge \neg(\exists z)(x < z \wedge z < y))$, says that q holds at every even point on the trace. In effect, Church was proposing to use classical logic (FO or MSO) as a logic of time, by focusing on infinite word structures. The set of infinite models of an FO or MSO sentence φ is denoted by $\text{models}_\omega(\varphi)$.

Church posed two problems related to sequential circuits [16]:

- The Decision problem: Given circuit C and a sentence φ, does φ hold in all traces of C? That is, does $\text{traces}(C) \subseteq \text{models}(\varphi)$ hold?
- The Synthesis problem: Given sets I and O of input and output signals, and a sentence φ over the vocabulary $I \cup O$, construct, if possible, a sequential circuit C with input signals I and output signals O such that φ holds in all traces of C. That is, construct C such that $\text{traces}(C) \subseteq \text{models}(\varphi)$ holds.

In modern terminology, Church's Decision problem is the model-checking problem in the linear-time approach (see Section 2.2). This problem did not receive much attention after [16,18], until the introduction of model checking in the early 1980s. In contrast, the Synthesis problem has remained a subject of ongoing research; see [19,20,21,22,23]. One reason that the Decision problem did not remain a subject of study, is the easy observation in [18] that the Decision problem can be reduced to the validity problem in the underlying logic (FO or MSO). Given a sequential circuit C, we can easily generate an FO sentence α_C that holds in precisely all structures associated with traces of C. Intuitively, the sentence α_C simply has to encode the transition and output functions of C, which are Boolean functions. Then φ holds in all traces of C precisely when $\alpha_C \rightarrow \varphi$ holds in all word structures (of the appropriate vocabulary). Thus, to solve the Decision problem we need to solve the Validity problem over word structures. As we see next, this problem was solved in 1962.

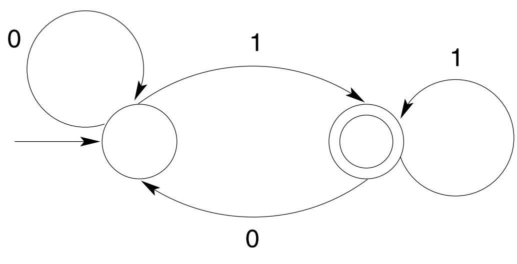
[2] We overload notation here and treat p as both a circuit element and a predicate symbol.

1.4 Reasoning about Infinite Words

Church's DECISION problem was essentially solved in 1962 by Büchi who showed that the VALIDITY problem over infinite word structures is decidable [24]. Actually, Büchi showed the decidability of the dual problem, which is the SATISFIABILITY problem for MSO over infinite word structures. Büchi's approach consisted of extending the automata-theoretic approach, see Theorem 1, which was introduced a few years earlier for word structures, to infinite word structures. To that end, Büchi extended automata theory to automata on infinite words.

A *nondeterministic Büchi automaton on words* (NBW) $A = (\Sigma, S, S_0, \rho, F)$ consists of a finite input alphabet Σ, a finite state set S, an initial state set $S_0 \subseteq S$, a transition relation $\rho \subseteq S \times \Sigma \times S$, and an accepting state set $F \subseteq S$. An NBW runs over an infinite input word $w = a_0, a_1, \ldots \in \Sigma^\omega$. A *run* of A on w is an infinite sequence $r = s_0, s_1, \ldots$ of states in S such that $s_0 \in S_0$, and $(s_i, a_i, s_{i+1}) \in \rho$, for $i \geq 0$. The run r is *accepting* if F is visited by r infinitely often; that is, $s_i \in F$ for infinitely many i's. The word w is *accepted* by A if A has an accepting run on w. The *infinitary language* of A, denoted $L_\omega(A)$, is the set of infinite words accepted by A. The class of languages accepted by NBWs forms the class of *ω-regular* languages, which are defined in terms of regular expressions augmented with the ω-power operator (e^ω denotes an infinitary iteration of e) [24].

Example 2. We describe graphically an NBW that accepts all words over the alphabet $\{0, 1\}$ that contain infinitely many occurrences of 1. The arrow on the left designates the initial state, and the circle on the right designates an accepting state. Note that this NBW looks exactly like the NFW in Example 1. The only difference is that in Example 1 we considered finite input words and here we are considering infinite input words.



As we saw earlier, the paradigmatic idea of the automata-theoretic approach is that we can compile high-level logical specifications into an equivalent low-level finite-state formalism.

Theorem 2. [24] *Given an MSO sentence φ with vocabulary V, one can construct an NBW A_φ with alphabet 2^V such that a word w in $(2^V)^\omega$ is accepted by A_φ iff φ holds in the word structure M_w. Conversely, given an NBW A with alphabet 2^V, one can construct an MSO sentence φ_A with vocabulary V such that φ_A holds in an infinite word structure M_w iff w is accepted by A.*

Thus, the class of languages defined by MSO sentences is precisely the class of ω-regular languages.

To decide whether sentence φ is satisfiable over infinite words, that is, whether $\text{models}_\omega(\varphi) \neq \emptyset$, we need to check that $L_\omega(A_\varphi) \neq \emptyset$. Let $A = (\Sigma, S, S_0, \rho, F)$ be an NBW. As with NFWs, construct a directed graph $G_A = (S, E_A)$, with S as the set of nodes, and $E_A = \{(s,t) : (s,a,t) \in \rho \text{ for some } a \in \Sigma\}$. The following lemma is implicit in [24] and more explicit in [25].

Lemma 2. *$L_\omega(A) \neq \emptyset$ iff there are states $s_0 \in S^0$ and $t \in F$ such that in G_A there is a path from s_0 to t and a path from t to itself.*

We thus obtain an algorithm for the Satisfiability problem of MSO over infinite word structures: given an MSO sentence φ, construct the NBW A_φ and check whether $L_\omega(A) \neq \emptyset$ by finding a path from an initial state to an accepting state and a cycle through that accepting state. Since the Decision problem can be reduced to the Satisfiability problem, this also solves the Decision problem.

Neither Büchi nor Church analyzed the complexity of the Decision problem. The non-elementary lower bound mentioned earlier for MSO over words can be easily extended to infinite words. The upper bound here is a bit more subtle. For both finite and infinite words, the construction of A_φ proceeds by induction on the structure of φ, with complementation being the difficult step. For NFW, complementation uses the *subset construction*, which involves a blow-up of 2^n [13,26]. Complementation for NBW is significantly more involved, see [27]. The blow-up of complementation is $2^{\Theta(n \log n)}$, but there is still a gap between the known upper and lower bounds. At any rate, this yields a blow-up of $exp(n, n \log n)$ for the translation from MSO to NBW.

2 Thread II: Temporal Logic

2.1 From Aristotle to Kamp

The history of time in logic goes back to ancient times.[3] Aristotle pondered how to interpret sentences such as "Tomorrow there will be a sea fight," or "Tomorrow there will not be a sea fight." Medieval philosophers also pondered the issue of time.[4] By the Renaissance period, philosophical interest in the logic

[3] For a detailed history of temporal logic from ancient times to the modern period, see [28].

[4] For example, William of Ockham, 1288–1348, wrote (rather obscurely for the modern reader): "Wherefore the difference between present tense propositions and past and future tense propositions is that the predicate in a present tense proposition stands in the same way as the subject, unless something added to it stops this; but in a past tense and a future tense proposition it varies, for the predicate does not merely stand for those things concerning which it is truly predicated in the past and future tense propositions, because in order for such a proposition to be true, it is not sufficient that that thing of which the predicate is truly predicated (whether by a verb in the present tense or in the future tense) is that which the subject denotes, although it is required that the very same predicate is truly predicated of that which the subject denotes, by means of what is asserted by such a proposition."

of time seems to have waned. There were some stirrings of interest in the 19th century, by Boole and Peirce. Peirce wrote:

> "Time has usually been considered by logicians to be what is called 'extra-logical' matter. I have never shared this opinion. But I have thought that logic had not yet reached the state of development at which the introduction of temporal modifications of its forms would not result in great confusion; and I am much of that way of thinking yet."

There were also some stirrings of interest in the first half of the 20th century, but the birth of modern temporal logic is unquestionably credited to Prior. Prior was a philosopher, who was interested in theological and ethical issues. His own religious path was somewhat convoluted; he was born a Methodist, converted to Presbytarianism, became an atheist, and ended up an agnostic. In 1949, he published a book titled "*Logic and The Basis of Ethics*". He was particularly interested in the conflict between the assumption of *free will* ("the future is to some extent, even if it is only a very small extent, something we can make for ourselves"), *foredestination* ("of what will be, it has now been the case that it will be"), and *foreknowledge* ("there is a deity who infallibly knows the entire future"). He was also interested in modal logic [29]. This confluence of interests led Prior to the development of *temporal logic.* [5] His wife, Mary Prior, recalled after his death:

> "I remember his waking me one night [in 1953], coming and sitting on my bed, ..., and saying he thought one could make a formalised tense logic."

Prior lectured on his new work when he was the John Locke Lecturer at the University of Oxford in 1955–6, and published his book "*Time and Modality*" in 1957 [31].[6] In this book, he presented a temporal logic that is propositional logic extended with two temporal connectives, F and P, corresponding to "sometime in the future" and "sometime in the past". A crucial feature of this logic is that it has an implicit notion of "now", which is treated as an *indexical*, that is, it depends on the context of utterance for its meaning. Both future and past are defined with respect to this implicit "now".

It is interesting to note that the *linear* vs. *branching* time dichotomy, which has been a subject of some controversy in the computer science literature since 1980 (see [32]), has been present from the very beginning of temporal-logic development. In Prior's early work on temporal logic, he assumed that time was linear. In 1958, he received a letter from Kripke,[7] who wrote

[5] An earlier term was *tense logic*; the term *temporal logic* was introduced in [30]. The technical distinction between the two terms seems fuzzy.

[6] Due to the arcane infix notation of the time, the book may not be too accessible to modern readers, who may have difficulties parsing formulas such as $CKMpMqAMKpMqMKqMp$.

[7] Kripke was a high-school student, not quite 18, in Omaha, Nebraska. Kripke's interest in modal logic was inspired by a paper by Prior on this subject [33]. Prior turned out to be the referee of Kripke's first paper [34].

> "In an indetermined system, we perhaps should not regard time as a linear series, as you have done. Given the present moment, there are several possibilities for what the next moment may be like – and for each possible next moment, there are several possibilities for the moment after that. Thus the situation takes the form, not of a linear sequence, but of a 'tree'."

Prior immediately saw the merit of Kripke's suggestion: "the determinist sees time as a line, and the indeterminist sees times as a system of forking paths." He went on to develop two theories of branching time, which he called "Ockhamist" and "Peircean". (Prior did not use path quantifiers; those were introduced later, in the 1980s. See Section 3.2.)

While the introduction of branching time seems quite reasonable in the context of trying to formalize free will, it is far from being simple philosophically. Prior argued that the nature of the course of time is branching, while the nature of a course of events is linear [35]. In contrast, it was argued in [30] that the nature of time is linear, but the nature of the course of events is branching: "We have 'branching *in* time,' not 'branching *of* time'."[8]

During the 1960s, the development of temporal logic continued through both the linear-time approach and the branching-time approach. There was little connection, however, between research on temporal logic and research on classical logics, as described in Section 1. That changed in 1968, when Kamp tied together the two threads in his doctoral dissertation.

Theorem 3. [36] *Linear temporal logic with past and binary temporal connectives ("strict until" and "strict since") has precisely the expressive power of FO over the ordered naturals (with monadic vocabularies).*

It should be noted that Kamp's Theorem is actually more general and asserts expressive equivalence of FO and temporal logic over all "Dedekind-closed orders". The introduction of binary temporal connectives by Kamp was necessary for reaching the expressive power of FO; *unary* linear temporal logic, which has only unary temporal connectives, is weaker than FO [37]. The theorem refers to FO formulas with one free variable, which are satisfied at an element of a structure, analogously to temporal logic formulas, which are satisfied at a point of time.

It should be noted that one direction of Kamp's Theorem, the translation from temporal logic to FO, is quite straightforward; the hard direction is the translation from FO to temporal logic. Both directions are algorithmically effective; translating from temporal logic to FO involves a linear blowup, but translation in the other direction involves a nonelementary blowup.

If we focus on FO sentences rather than FO formulas, then they define sets of traces (a sentence φ defines models(φ)). A characterization of of the

[8] One is reminded of St. Augustin, who said in his *Confessions*: "What, then, is time? If no one asks me, I know; but if I wish to explain it to some who should ask me, I do not know."

expressiveness of FO sentences over the naturals, in terms of their ability to define sets of traces, was obtained in 1979.

Theorem 4. [38] *FO sentences over naturals have the expressive power of $*$-free ω-regular expressions.*

Recall that MSO defines the class of ω-regular languages. It was already shown in [39] that FO over the naturals is weaker expressively than MSO over the naturals. Theorem 4 was inspired by an analogous theorem in [40] for finite words.

2.2 The Temporal Logic of Programs

There were some early observations that temporal logic can be applied to programs. Prior stated: "There are practical gains to be had from this study too, for example, in the representation of time-delay in computer circuits" [35]. Also, a discussion of the application of temporal logic to processes, which are defined as "programmed sequences of states, deterministic or stochastic" appeared in [30].

The "big bang" for the application of temporal logic to program correctness occurred with Pnueli's 1977 paper [41]. In this paper, Pnueli, inspired by [30], advocated using future linear temporal logic (LTL) as a logic for the specification of non-terminating programs; see overview in [42].

LTL is a temporal logic with two temporal connectives, "next" and "until".[9] In LTL, formulas are constructed from a set *Prop* of atomic propositions using the usual Boolean connectives as well as the unary temporal connective X ("next"), and the binary temporal connective U ("until"). Additional unary temporal connectives F ("eventually"), and G ("always") can be defined in terms of U. Note that all temporal connectives refer to the future here, in contrast to Kamp's "strict since" operator, which refers to the past. Thus, LTL is a *future temporal logic*. For extensions with past temporal connectives, see [43,44,45].

LTL is interpreted over traces over the set *Prop* of atomic propositions. For a trace τ and a point $i \in \mathrm{N}$, the notation $\tau, i \models \varphi$ indicates that the formula φ holds at the point i of the trace τ. Thus, the point i is the implicit "now" with respect to which the formula is interpreted. We have that

- $\tau, i \models p$ if p holds at $\tau(i)$,
- $\tau, i \models X\varphi$ if $\tau, i+1 \models \varphi$, and
- $\tau, i \models \varphi U \psi$ if for some $j \geq i$, we have $\tau, j \models \psi$ and for all k, $i \leq k < j$, we have $\tau, k \models \varphi$.

The temporal connectives F and G can be defined in terms of the temporal connective U; $F\varphi$ is defined as **true** $U\varphi$, and $G\varphi$ is defined as $\neg F \neg \varphi$. We say that τ *satisfies* a formula φ, denoted $\tau \models \varphi$, iff $\tau, 0 \models \varphi$. We denote by models(φ) the set of traces satisfying φ.

[9] Unlike Kamp's "strict until" ("p strict until q" requires q to hold in the strict future), Pnueli's "until" is not strict ("p until q" can be satisfied by q holding now), which is why the "next" connective is required.

As an example, the LTL formula $G(request \rightarrow F\ grant)$, which refers to the atomic propositions *request* and *grant*, is true in a trace precisely when every state in the trace in which *request* holds is followed by some state in the (non-strict) future in which *grant* holds. Also, the LTL formula $G(request \rightarrow (request\ U\ grant))$ is true in a trace precisely if, whenever *request* holds in a state of the trace, it holds until a state in which *grant* holds is reached.

The focus on satisfaction at 0, called *initial semantics*, is motivated by the desire to specify computations at their starting point. It enables an alternative version of Kamp's Theorem, which does not require past temporal connectives, but focuses on initial semantics.

Theorem 5. [46] *LTL has precisely the expressive power of FO over the ordered naturals (with monadic vocabularies) with respect to initial semantics.*

As we saw earlier, FO has the expressive power of star-free ω-regular expressions over the naturals. Thus, LTL has the expressive power of star-free ω-regular expressions (see [47]), and is strictly weaker than MSO. An interesting outcome of the above theorem is that it lead to the following assertion regarding LTL [48]: "The corollary due to Meyer – I have to get in my controversial remark – is that that [Theorem 5] makes it theoretically uninteresting." Developments since 1980 have proven this assertion to be overly pessimistic on the merits of LTL.

Pnueli also discussed the analog of Church's DECISION problem: given a finite-state program P and an LTL formula φ, decide if φ holds in all traces of P. Just like Church, Pnueli observed that this problem can be solved by reduction to MSO. Rather than focus on sequential circuits, Pnueli focused on programs, modeled as (labeled) *transition systems* [49]. A transition system $M = (W, W_0, R, V)$ consists of a set W of states that the system can be in, a set $W_0 \subseteq W$ of initial states, a transition relation $R \subseteq W^2$ that indicates the allowable state transitions of the system, and an assignment $V : W \rightarrow 2^{Prop}$ of truth values to the atomic propositions in each state of the system. (A transition system is essentially a Kripke structure [50].) A *path* in M that *starts at* u is a possible infinite behavior of the system starting at u, i.e., it is an infinite sequence $u_0, u_1 \ldots$ of states in W such that $u_0 = u$, and $(u_i, u_{i+1}) \in R$ for all $i \geq 0$. The sequence $V(u_0), V(u_1) \ldots$ is a *trace* of M that *starts at* u. It is the sequence of truth assignments visited by the path. The *language* of M, denoted $L(M)$, consists of all traces of M that start at a state in W_0. Note that $L(M)$ is a language of infinite words over the alphabet 2^{Prop}. The language $L(M)$ can be viewed as an abstract description of the system M, describing all possible traces. We say that M *satisfies* an LTL formula φ if all traces in $L(M)$ satisfy φ, that is, if $L(M) \subseteq \text{models}(\varphi)$. When W is finite, we have a finite-state system, and can apply algorithmic techniques.

What about the complexity of LTL reasoning? Recall from Section 1 that satisfiability of FO over trace structures is nonelementary. In contrast, it was shown in [51,52,53,54,55,56,57] that LTL SATISFIABILITY is elementary; in fact, it is PSPACE-complete. It was also shown that the DECISION problem for LTL

with respect to finite transition systems is PSPACE-complete [53,54,55]. The basic technique for proving these elementary upper bounds is the *tableau* technique, which was adapted from *dynamic logics* [58] (see Section 3.1). Thus, even though FO and LTL are expressively equivalent, they have dramatically different computational properties, as LTL reasoning is in PSPACE, while FO reasoning is nonelementary.

The second "big bang" in the application of temporal logic to program correctness was the introduction of *model checking* by Clarke and Emerson [59] and by Queille and Sifakis [60]. The two papers used two different branching-time logics. Clarke and Emerson used CTL (inspired by the branching-time logic UB of [61]), which extends LTL with existential and universal path quantifiers E and A. Queille and Sifakis used a logic introduced by Leslie Lamport [17], which extends propositional logic with the temporal connectives POT (which corresponds to the CTL operator EF) and $INEV$ (which corresponds to the CTL operator AF). The focus in both papers was on model checking, which is essentially what Church called the DECISION problem: does a given finite-state program, viewed as a finite transition system, satisfy its given temporal specification. In particular, Clarke and Emerson showed that model checking transition systems of size m with respect to formulas of size n can be done in time polynomial in m and n. This was refined later to $O(mn)$ (even in the presence of *fairness* constraints, which restrict attention to certain infinite paths in the underlying transition system) [62,63]. We drop the term "DECISION problem" from now on, and replace it with the term "MODEL-CHECKING problem".[10]

It should be noted that the linear complexity of model checking refers to the size of the transition system, rather than the size of the program that gave rise to that system. For sequential circuits, transition-system size is essentially exponential in the size of the description of the circuit (say, in some Hardware Description Language). This is referred to as the "state-explosion problem" [65]. In spite of the state-explosion problem, in the first few years after the publication of the first model-checking papers in 1981-2, Clarke and his students demonstrated that model checking is a highly successful technique for automated program verification [66,67]. By the late 1980s, automated verification had become a recognized research area. Also by the late 1980s, *symbolic* model checking was developed [68,69], and the SMV tool, developed at CMU by McMillan [70], was starting to have an industrial impact. See [71] for more details.

The detailed complexity analysis in [62] inspired a similar detailed analysis of linear time model checking. It was shown in [72] that model checking transition systems of size m with respect to LTL formulas of size n can be done in time $m2^{O(n)}$. (This again was shown using a tableau-based technique.) While the bound here is exponential in n, the argument was that n is typically rather small, and therefore an exponential bound is acceptable.

[10] The model-checking problem is analogous to database query evaluation, where we check the truth of a logical formula, representing a query, with respect to a database, viewed as a finite relational structure. Interestingly, the study of the complexity of database query evaluation started about the same time as that of model checking [64].

2.3 Back to Automata

Since LTL can be translated to FO, and FO can be translated to NBW, it is clear that LTL can be translated to NBW. Going through FO, however, would incur, in general, a nonelementary blowup. In 1983, Wolper, Sistla, and I showed that this nonelementary blowup can be avoided.

Theorem 6. [73,74] *Given an LTL formula φ of size n, one can construct an NBW A_φ of size $2^{O(n)}$ such that a trace σ satisfies φ if and only if σ is accepted by A_φ.*

It now follows that we can obtain a PSPACE algorithm for LTL SATISFIABILITY: given an LTL formula φ, we construct A_φ and check that $A_\varphi \neq \emptyset$ using the graph-theoretic approach described earlier. We can avoid using exponential space, by constructing the automaton *on the fly* [73,74].

What about model checking? We know that a transition system M satisfies an LTL formula φ if $L(M) \subseteq \text{models}(\varphi)$. It was then observed in [75] that the following are equivalent:

- M satisfies φ
- $L(M) \subseteq \text{models}(\varphi)$
- $L(M) \subseteq L(A_\varphi)$
- $L(M) \cap ((2^{Prop})^\omega - L(A_\varphi)) = \emptyset$
- $L(M) \cap L(A_{\neg\varphi}) = \emptyset$
- $L(M \times A_{\neg\varphi}) = \emptyset$

Thus, rather than complementing A_φ using an exponential complementation construction [24,76,77], we complement the LTL property using logical negation. It is easy to see that we can now get the same bound as in [72]: model checking programs of size m with respect to LTL formulas of size n can be done in time $m2^{O(n)}$. Thus, the optimal bounds for LTL satisfiability and model checking can be obtained without resorting to ad-hoc tableau-based techniques; the key is the exponential translation of LTL to NBW.

One may wonder whether this theory is practical. Reduction to practice took over a decade of further research, which saw the development of

- an optimized search algorithm for explicit-state model checking [78,79],
- a symbolic, BDD-based[11] algorithm for NBW nonemptiness [68,69,81],
- symbolic algorithms for LTL to NBW translation [68,69,82], and
- an optimized explicit algorithm for LTL to NBW translation [83].

By 1995, there were two model-checking tools that implemented LTL model checking via the automata-theoretic approach: Spin [84] is an explicit-state LTL model checker, and Cadence's SMV is a symbolic LTL model checker.[12] See [85] for a description of algorithmic developments since the mid 1990s. Additional tools today are *VIS* [86], *NuSMV* [87], and *SPOT* [88].

[11] To be precise, one should use the acronym ROBDD, for Reduced Ordered Binary Decision Diagrams [80].

[12] Cadence's SMV is also a CTL model checker. See `www.cadence.com/webforms/cbl_software/index.aspx`.

It should be noted that Kurshan developed the automata-theoretic approach independently, also going back to the 1980s [89,90,91]. In his approach (as also in [92,74]), one uses automata to represent both the system and its specification [93].[13] The first implementation of COSPAN, a model-checking tool that is based on this approach [94], also goes back to the 1980s; see [95].

2.4 Enhancing Expressiveness

Can the development of LTL model checking [72,75] be viewed as a satisfactory solution to Church's DECISION problem? Almost, but not quite, since, as we observed earlier, LTL is not as expressive as MSO, which means that LTL is expressively weaker than NBW. Why do we need the expressive power of NBWs? First, note that once we add fairness to transitions systems (sse [62,63]), they can be viewed as variants of NBWs. Second, there are good reasons to expect the specification language to be as expressive as the underlying model of programs [96]. Thus, achieving the expressive power of NBWs, which we refer to as ω-*regularity*, is a desirable goal. This motivated efforts since the early 1980s to extend LTL.

The first attempt along this line was made by Wolper [56,57], who defined ETL (for *Extended Temporal Logic*), which is LTL extended with grammar operators. He showed that ETL is more expressive than LTL, while its SATISFIABILITY problem can still be solved in exponential time (and even PSPACE [53,54,55]). Then, Sistla, Wolper and I showed how to extend LTL with automata connectives, reaching ω-regularity, without losing the PSPACE upper bound for the SATISFIABILITY problem [73,74]. Actually, three syntactical variations, denoted ETL_f, ETL_l, and ETL_r were shown to be expressively equivalent and have these properties [73,74].

Two other ways to achieve ω-regularity were discovered in the 1980s. The first is to enhance LTL with monadic second-order quantifiers as in MSO, which yields a logic, QPTL, with a nonelementary SATISFIABILITY problem [97,77]. The second is to enhance LTL with least and greatest fixpoints [98,99], which yields a logic, μLTL, that achieves ω-regularity, and has a PSPACE upper bound on its SATISFIABILITY and MODEL-CHECKING problems [99]. For example, the (not too readable) formula

$$(\nu P)(\mu Q)(P \wedge X(p \vee Q)),$$

where ν and μ denote greatest and least fixpoint operators, respectively, is equivalent to the LTL formula GFp, which says that p holds infinitely often.

3 Thread III: Dynamic and Branching-Time Logics

3.1 Dynamic Logics

In 1976, a year before Pnueli proposed using LTL to specify programs, Pratt proposed using *dynamic logic*, an extension of modal logic, to specify programs

[13] The connection to automata is somewhat difficult to discern in the early papers [89,90].

[100].[14] In modal logic $\Box\varphi$ means that φ holds in all worlds that are possible with respect to the current world [50]. Thus, $\Box\varphi$ can be taken to mean that φ holds after an execution of a program step, taking the transition relation of the program to be the possibility relation of a Kripke structure. Pratt proposed the addition of dynamic modalities $[e]\varphi$, where e is a program, which asserts that φ holds in all states reachable by an execution of the program e. Dynamic logic can then be viewed as an extension of Hoare logic, since $\psi \rightarrow [e]\varphi$ corresponds to the Hoare triple $\{\psi\}e\{\varphi\}$ (see [106]). See [105] for an extensive coverage of dynamic logic.

In 1977, a propositional version of Pratt's dynamic logic, called PDL, was proposed, in which programs are regular expressions over atomic programs [107,108]. It was shown there that the SATISFIABILITY problem for PDL is in NEXPTIME and EXPTIME-hard. Pratt then proved an EXPTIME upper bound, adapting tableau techniques from modal logic [58,109]. (We saw earlier that Wolper then adapted these techniques to linear-time logic.)

Pratt's dynamic logic was designed for terminating programs, while Pnueli was interested in nonterminating programs. This motivated various extensions of dynamic logic to nonterminating programs [110,111,112,113]. Nevertheless, these logics are much less natural for the specification of ongoing behavior than temporal logic. They inspired, however, the introduction of the (*modal*) *μ-calculus* by Kozen [114,115]. The μ-calculus is an extension of modal logic with least and greatest fixpoints. It subsumes expressively essentially all dynamic and temporal logics [116]. Kozen's paper was inspired by previous papers that showed the usefulness of fixpoints in characterizing correctness properties of programs [117,118] (see also [119]). In turn, the μ-calculus inspired the introduction of μLTL, mentioned earlier. The μ-calculus also played an important role in the development of symbolic model checking [68,69,81].

3.2 Branching-Time Logics

Dynamic logic provided a branching-time approach to reasoning about programs, in contrast to Pnueli's linear-time approach. Lamport was the first to study the dichotomy between linear and branching time in the context of program correctness [17]. This was followed by the introduction of the branching-time logic UB, which extends unary LTL (LTL without the temporal connective "until") with the existential and universal path quantifiers, E and A [61]. Path quantifiers enable us to quantify over different future behavior of the system. By adapting Pratt's tableau-based method for PDL to UB, it was shown that its SATISFIABILITY problem is in EXPTIME [61]. Clarke and Emerson then added the temporal conncetive "until" to UB and obtained CTL [59]. (They did not focus on the SATISFIABILITY problem for CTL, but, as we saw earlier, on its MODEL-CHECKING problem; the SATISFIABILITY problem was shown later to be solvable in EXPTIME [120].) Finally, it was shown that LTL and CTL have incomparable expressive power, leading to the introduction of the branching-time logic CTL*, which unifies LTL and CTL [121,122].

[14] See discussion of precursor and related developments, such as [101,102,103,104], in [105].

The key feature of branching-time logics in the 1980s was the introduction of explicit path quantifiers in [61]. This was an idea that was not discovered by Prior and his followers in the 1960s and 1970s. Most likely, Prior would have found CTL* satisfactory for his philosophical applications and would have seen no need to introduce the "Ockhamist" and "Peircean" approaches.

3.3 Combining Dynamic and Temporal Logics

By the early 1980s it became clear that temporal logics and dynamic logics provide two distinct perspectives for specifying programs: the first is *state* based, while the second is *action* based. Various efforts have been made to combine the two approaches. These include the introduction of *Process Logic* [123] (branching time), *Yet Another Process Logic* [124] (branching time), *Regular Process Logic* [125] (linear time), *Dynamic LTL* [126] (linear time), and *RCTL* [127] (branching time), which ultimately evolved into *Sugar* [128]. RCTL/Sugar is unique among these logics in that it did not attempt to borrow the action-based part of dynamic logic. It is a state-based branching-time logic with no notion of actions. Rather, what it borrowed from dynamic logic was the use of regular-expression-based dynamic modalities. Unlike dynamic logic, which uses regular expressions over program statements, RCTL/Sugar uses regular expressions over state predicates, analogously to the automata of ETL [73,74], which run over sequences of formulas.

4 Thread IV: From LTL to ForSpec, PSL, and SVA

In the late 1990s and early 2000s, model checking was having an increasing industrial impact. That led to the development of three industrial temporal logics based on LTL: *ForSpec*, developed by Intel, and *PSL* and *SVA*, developed by industrial standards committees.

4.1 From LTL to ForSpec

Intel's involvement with model checking started in 1990, when Kurshan, spending a sabbatical year in Israel, conducted a successful feasibility study at the Intel Design Center (IDC) in Haifa, using COSPAN, which at that point was a prototype tool; see [95]. In 1992, IDC started a pilot project using SMV. By 1995, model checking was used by several design projects at Intel, using an internally developed model checker based on SMV. Intel users have found CTL to be lacking in expressive power and the Design Technology group at Intel developed its own specification language, FSL. The FSL language was a linear-time logic, and it was model checked using the automata-theoretic approach, but its design was rather ad-hoc, and its expressive power was unclear; see [129].

In 1997, Intel's Design Technology group at IDC embarked on the development of a second-generation model-checking technology. The goal was to develop a model-checking engine from scratch, as well as a new specification language. A BDD-based model checker was released in 1999 [130], and a SAT-based model checker was released in 2000 [131].

I got involved in the design of the second-generation specification language in 1997. That language, ForSpec, was released in 2000 [132]. The first issue to be decided was whether the language should be linear or branching. This led to an in-depth examination of this issue [32], and the decision was to pursue a linear-time language. An obvious candidate was LTL; we saw that by the mid 1990s there were both explicit-state and symbolic model checkers for LTL, so there was no question of feasibility. I had numerous conversations with L. Fix, M. Hadash, Y. Kesten, and M. Sananes on this issue. The conclusion was that LTL is not expressive enough for industrial usage. In particular, many properties that are expressible in FSL are not expressible in LTL. Thus, it turned out that the theoretical considerations regarding the expressiveness of LTL, i.e., its lack of ω-regularity, had practical significance. I offered two extensions of LTL; as we saw earlier both ETL and μLTL achieve ω-regularity and have the same complexity as LTL. Neither of these proposals was accepted, due to the perceived difficulty of usage of such logics by Intel validation engineers, who typically have only basic familiarity with automata theory and logic.

These conversations continued in 1998, now with A. Landver. Avner also argued that Intel validation engineers would not be receptive to the automata-based formalism of ETL. Being familiar with RCTL/Sugar and its dynamic modalities [128,127], he asked me about regular expressions, and my answer was that regular expressions are equivalent to automata [6], so the automata of ETL_f, which extends LTL with automata on *finite* words, can be replaced by regular expressions over state predicates. This lead to the development of *RELTL*, which is LTL augmented by the dynamic regular modalities of dynamic logic (interpreted linearly, as in ETL). Instead of the dynamic-logic notation $[e]\varphi$, ForSpec uses the more readable (to engineers) (e $\mathsf{triggers}$ φ), where e is a regular expression over state predicates (e.g., $(p \vee q)^*, (p \wedge q)$), and φ is a formula. Semantically, $\tau, i \models (e\ \mathsf{triggers}\ \varphi)$ if, for all $j \geq i$, if $\tau[i,j]$ (that is, the finite word $\tau(i), \ldots, \tau(j)$) "matches" e (in the intuitive formal sense), then $\tau, j \models \varphi$; see [133]. Using the ω-regularity of ETL_f, it is now easy to show that RELTL also achieves ω-regularity [132].

While the addition of dynamic modalities to LTL is sufficient to achieve ω-regularity, we decided to also offer direct support to two specification modes often used by verification engineers at Intel: *clocks* and *resets*. Both clocks and resets are features that are needed to address the fact that modern semiconductor designs consist of interacting parallel modules. While clocks and resets have a simple underlying intuition, defining their semantics formally is quite nontrivial. ForSpec is essentially RELTL, augmented with features corresponding to clocks and resets, as we now explain.

Today's semiconductor designs are still dominated by synchronous circuits. In synchronous circuits, clock signals synchronize the sequential logic, providing the designer with a simple operational model. While the asynchronous approach holds the promise of greater speed (see [134]), designing asynchronous circuits is significantly harder than designing synchronous circuits. Current design methodology attempts to strike a compromise between the two approaches by using

multiple clocks. This results in architectures that are globally asynchronous but locally synchronous. The temporal-logic literature mostly ignores the issue of explicitly supporting clocks. ForSpec supports multiple clocks via the notion of *current clock*. Specifically, ForSpec has a construct change_on $c\ \varphi$, which states that the temporal formula φ is to be evaluated with respect to the clock c; that is, the formula φ is to be evaluated in the trace defined by the high phases of the clock c. The key feature of clocks in ForSpec is that each subformula may advance according to a different clock [132].

Another feature of modern designs' consisting of interacting parallel modules is the fact that a process running on one module can be reset by a signal coming from another module. As noted in [135], reset control has long been a critical aspect of embedded control design. ForSpec directly supports reset signals. The formula accept_on $a\ \varphi$ states that the property φ should be checked only until the arrival of the reset signal a, at which point the check is considered to have *succeeded*. In contrast, reject_on $r\ \varphi$ states that the property φ should be checked only until the arrival of the reset signal r, at which point the check is considered to have *failed*. The key feature of resets in ForSpec is that each subformula may be reset (positively or negatively) by a different reset signal; for a longer discussion see [132].

ForSpec is an industrial property-specification language that supports hardware-oriented constructs as well as uniform semantics for formal and dynamic validation, while at the same time it has a well understood expressiveness (ω-regularity) and computational complexity (SATISFIABILITY and MODEL-CHECKING problems have the same complexity for ForSpec as for LTL) [132]. The design effort strove to find an acceptable compromise, with trade-offs clarified by theory, between conflicting demands, such as expressiveness, usability, and implementability. Clocks and resets, both important to hardware designers, have a clear intuitive semantics, but formalizing this semantics is nontrivial. The rigorous semantics, however, not only enabled mechanical verification of various theorems about the language, but also served as a reference document for the implementors. The implementation of model checking for ForSpec followed the automata-theoretic approach, using *alternating* automata as advocated in [136] (see [137]).

4.2 From ForSpec to PSL and SVA

In 2000, the Electronic Design Automation Association instituted a standardization body called *Accellera*.[15] Accellera's mission is to drive worldwide development and use of standards required by systems, semiconductor and design tools companies. Accellera decided that the development of a standard specification language is a requirement for formal verification to become an industrial reality (see [95]). Since the focus was on specifying properties of designs rather than designs themselves, the chosen term was "property specification language" (PSL). The PSL standard committee solicited industrial contributions and received four language contributions: *CBV*, from Motorola, ForSpec, from Intel, *Temporal e*, from Verisity [138], and Sugar, from IBM.

[15] See `http://www.accellera.org/`

The committee's discussions were quite fierce.[16] Ultimately, it became clear that while technical considerations play an important role, industrial committees' decisions are ultimately made for business considerations. In that contention, IBM had the upper hand, and Accellera chose Sugar as the base language for PSL in 2003. At the same time, the technical merits of ForSpec were accepted and PSL adopted all the main features of ForSpec. In essence, PSL (the current version 1.1) is LTL, extended with dynamic modalities (referred to as the *regular layer*), clocks, and resets (called *aborts*). PSL did inherit the syntax of Sugar, and does include a branching-time extension as an acknowledgment to Sugar.[17]

There was some evolution of PSL with respect to ForSpec. After some debate on the proper way to define resets [140], ForSpec's approach was essentially accepted after some reformulation [141]. ForSpec's fundamental approach to clocks, which is semantic, was accepted, but modified in some important details [142]. In addition to the dynamic modalities, borrowed from dynamic logic, PSL also has weak dynamic modalities [143], which are reminiscent of "looping" modalities in dynamic logic [110,144]. Today PSL 1.1 is an IEEE Standard 1850–2005, and continues to be refined by the IEEE P1850 PSL Working Group.[18]

Practical use of ForSpec and PSL has shown that the regular layer (that is, the dynamic modalities), is highly popular with verification engineers. Another standardized property specification language, called *SVA* (for SystemVerilog Assertions), is based, in essence, on that regular layer [145].

5 Contemplation

This evolution of ideas, from Löwenheim and Skolem to PSL and SVA, seems to me to be an amazing development. It reminds me of the medieval period, when building a cathedral spanned more than a mason's lifetime. Many masons spend their whole lives working on a cathedral, never seeing it to completion. We are fortunate to see the completion of this particular "cathedral". Just like the medieval masons, our contributions are often smaller than we'd like to consider them, but even small contributions can have a major impact. Unlike the medieval cathedrals, the scientific cathedral has no architect; the construction is driven by a complex process, whose outcome is unpredictable. Much that has been discovered is forgotten and has to be rediscovered. It is hard to fathom what our particular "cathedral" will look like in 50 years.

Acknowledgments

I am grateful to E. Clarke, A. Emerson, R. Goldblatt, A. Pnueli, P. Sistla, P. Wolper for helping me trace the many threads of this story, to D. Fisman,

[16] See `http://www.eda-stds.org/vfv/`

[17] See [139] and language reference manual at `http://www.eda.org/vfv/docs/PSL-v1.1.pdf`

[18] See `http://www.eda.org/ieee-1850/`

C. Eisner, J. Halpern, D. Harel and T. Wilke for their many useful comments on earlier drafts of this paper, and to S. Nain, K. Rozier, and D. Tabakov for proofreading earlier drafts. I'd also like to thank K. Rozier for her help with graphics.

References

1. Davis, M.: Engines of Logic: Mathematicians and the Origin of the Computer. Norton (2001)
2. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. Springer, Heidelberg (1996)
3. Dreben, D., Goldfarb, W.D.: The Decision Problem: Solvable Classes of Quantificational Formulas. Addison-Wesley, Reading (1979)
4. Löwenheim, L.: Über Möglichkeiten im Relativkalküll (On possibilities in the claculus of relations). Math. Ann. 76, 447–470 (1915); Translated in From Frege to Gödel, van Heijenoort. Harvard Univ. Press (1971)
5. Skolem, T.: Untersuchung über Axiome des Klassenkalküls und über Produktations- und Summationsprobleme, welche gewisse Klassen von Aussagen betreffen (Investigations of the axioms of the calculusof classes and on product and sum problems that are connected with certain class of statements). Videnskabsakademiet i Kristiania, Skrifter I 3 (1919); Translated in Selected Works in Logic by Th. Skolem", Fenstak, J.E., Scand. Univ. Books, Universitetsforlaget, Oslo, 67–101(1970)
6. Hopcroft, J., Ullman, J.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979)
7. Büchi, J.: Weak second-order arithmetic and finite automata. Zeit. Math. Logik und Grundl. Math. 6, 66–92 (1960)
8. Büchi, J., Elgot, C., Wright, J.: The non-existence of certain algorithms for finite automata theory (abstract). Notices Amer. Math. Soc. 5, 98 (1958)
9. Elgot, C.: Decision problems of finite-automata design and related arithmetics. Trans. Amer. Math. Soc. 98, 21–51 (1961)
10. Trakhtenbrot, B.: The synthesis of logical nets whose operators are described in terms of one-place predicate calculus. Doklady Akad. Nauk SSSR 118(4), 646–649 (1958)
11. Trakhtenbrot, B.: Certain constructions in the logic of one-place predicates. Doklady Akad. Nauk SSSR 138, 320–321 (1961)
12. Trakhtenbrot, B.: Finite automata and monadic second order logic. Siberian Math. J. 3, 101–131 (1962) (in Russian); English translation In: AMS Transl. 59, 23–55 (1966)
13. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Journal of Research and Development 3, 115–125 (1959)
14. Meyer, A.R.: Weak monadic second order theory of successor is not elementary recursive. In: Proc. Logic Colloquium. Lecture Notes in Mathematics, vol. 453, pp. 132–154. Springer, Heidelberg (1975)
15. Stockmeyer, L.: The complexity of decision procedures in Automata Theory and Logic. PhD thesis, MIT, Project MAC Technical Report TR-133 (1974)
16. Church, A.: Applicaton of recursive arithmetics to the problem of circuit synthesis. In: The Summer Institute for Symbolic Logic, Communications Research Division, Institute for Defense Analysis, pp. 3–50 (1957)

17. Lamport, L.: “Sometimes” is sometimes “not never” - on the temporal logic of programs. In: Proc. 7th ACM Symp. on Principles of Programming Languages, pp. 174–185 (1980)
18. Church, A.: Logic, arithmetics, and automata. In: Proc. Int. Congress of Mathematicians, 1962, Institut Mittag-Leffler, pp. 23–35 (1963)
19. Büchi, J., Landweber, L.: Solving sequential conditions by finite-state strategies. Trans. AMS 138, 295–311 (1969)
20. Kupferman, O., Piterman, N., Vardi, M.: Safraless compositional synthesis. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 31–44. Springer, Heidelberg (2006)
21. Kupferman, O., Vardi, M.: Safraless decision procedures. In: Proc. 46th IEEE Symp. on Foundations of Computer Science, pp. 531–540 (2005)
22. Rabin, M.: Automata on infinite objects and Church’s problem. Amer. Mathematical Society (1972)
23. Thomas, W.: On the synthesis of strategies in infinite games. In: Mayr, E., Puech, C. (eds.) STACS 1995. LNCS, vol. 900, pp. 1–13. Springer, Heidelberg (1995)
24. Büchi, J.: On a decision method in restricted second order arithmetic. In: Proc. Int. Congress on Logic, Method, and Philosophy of Science, 1960, pp. 1–12. Stanford University Press (1962)
25. Trakhtenbrot, B., Barzdin, Y.: Finite Automata. North Holland, Amsterdam (1973)
26. Sakoda, W., Sipser, M.: Non-determinism and the size of two-way automata. In: Proc. 10th ACM Symp. on Theory of Computing, pp. 275–286 (1978)
27. Vardi, M.Y.: The büchi complementation saga. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 12–22. Springer, Heidelberg (2007)
28. Øhrstrøm, P., Hasle, P.: Temporal Logic: from Ancient Times to Artificial Intelligence. In: Studies in Linguistics and Philosophy, vol. 57. Kluwer, Dordrecht (1995)
29. Prior, A.: Modality de dicto and modality de re. Theoria 18, 174–180 (1952)
30. Rescher, N., Urquhart, A.: Temporal Logic. Springer, Heidelberg (1971)
31. Prior, A.: Time and Modality. Oxford University Press, Oxford (1957)
32. Vardi, M.: Branching vs. linear time: Final showdown. In: Margaria, T., Yi, W. (eds.) TACAS 2001. LNCS, vol. 2031, pp. 1–22. Springer, Heidelberg (2001)
33. Prior, A.: Modality and quantification in s5. J. Symbolic Logic 21, 60–62 (1956)
34. Kripke, S.: A completeness theorem in modal logic. Journal of Symbolic Logic 24, 1–14 (1959)
35. Prior, A.: Past, Present, and Future. Clarendon Press (1967)
36. Kamp, J.: Tense Logic and the Theory of Order. PhD thesis, UCLA (1968)
37. Etessami, K., Vardi, M., Wilke, T.: First-order logic with two variables and unary temporal logic. Inf. Comput. 179(2), 279–295 (2002)
38. Thomas, W.: Star-free regular sets of ω-sequences. Information and Control 42(2), 148–156 (1979)
39. Elgot, C., Wright, J.: Quantifier elimination in a problem of logical design. Michigan Math. J. 6, 65–69 (1959)
40. McNaughton, R., Papert, S.: Counter-Free Automata. MIT Press, Cambridge (1971)
41. Pnueli, A.: The temporal logic of programs. In: Proc. 18th IEEE Symp. on Foundations of Computer Science, pp. 46–57 (1977)
42. Goldblatt, R.: Logic of time and computation. Technical report, CSLI Lecture Notes, no.7. Stanford University (1987)

43. Lichtenstein, O., Pnueli, A., Zuck, L.: The glory of the past. In: Parikh, R. (ed.) Logic of Programs 1985. LNCS, vol. 193, pp. 196–218. Springer, Heidelberg (1985)
44. Markey, N.: Temporal logic with past is exponentially more succinct. EATCS Bulletin 79, 122–128 (2003)
45. Vardi, M.: A temporal fixpoint calculus. In: Proc. 15th ACM Symp. on Principles of Programming Languages, pp. 250–259 (1988)
46. Gabbay, D., Pnueli, A., Shelah, S., Stavi, J.: On the temporal analysis of fairness. In: Proc. 7th ACM Symp. on Principles of Programming Languages, pp. 163–173 (1980)
47. Pnueli, A., Zuck, L.: In and out of temporal logic. In: Proc. 8th IEEE Symp. on Logic in Computer Science, pp. 124–135 (1993)
48. Meyer, A.: Ten thousand and one logics of programming. Technical report, MIT, MIT-LCS-TM-150 (1980)
49. Keller, R.: Formal verification of parallel programs. Communications of the ACM 19, 371–384 (1976)
50. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2002)
51. Halpern, J., Reif, J.: The propositional dynamic logic of deterministic, well-structured programs (extended abstract). In: Proc. 22nd IEEE Symp. on Foundations of Computer Science, pp. 322–334 (1981)
52. Halpern, J., Reif, J.: The propositional dynamic logic of deterministic, well-structured programs. Theor. Comput. Sci. 27, 127–165 (1983)
53. Sistla, A.: Theoretical issues in the design of distributed and concurrent systems. PhD thesis, Harvard University (1983)
54. Sistla, A., Clarke, E.: The complexity of propositional linear temporal logics. In: Proc. 14th Annual ACM Symposium on Theory of Computing, pp. 159–168 (1982)
55. Sistla, A., Clarke, E.: The complexity of propositional linear temporal logic. Journal of the ACM 32, 733–749 (1985)
56. Wolper, P.: Temporal logic can be more expressive. In: Proc. 22nd IEEE Symp. on Foundations of Computer Science, pp. 340–348 (1981)
57. Wolper, P.: Temporal logic can be more expressive. Information and Control 56(1–2), 72–99 (1983)
58. Pratt, V.: A near-optimal method for reasoning about action. Journal of Computer and Systems Science 20(2), 231–254 (1980)
59. Clarke, E., Emerson, E.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)
60. Queille, J., Sifakis, J.: Specification and verification of concurrent systems in Cesar. In: Proc. 9th ACM Symp. on Principles of Programming Languages. LNCS, vol. 137. Springer, Heidelberg (1982)
61. Ben-Ari, M., Manna, Z., Pnueli, A.: The logic of nexttime. In: Proc. 8th ACM Symp. on Principles of Programming Languages, pp. 164–176 (1981)
62. Clarke, E., Emerson, E., Sistla, A.: Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. In: Proc. 10th ACM Symp. on Principles of Programming Languages, pp. 117–126 (1983)
63. Clarke, E., Emerson, E., Sistla, A.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languagues and Systems 8(2), 244–263 (1986)
64. Vardi, M.: The complexity of relational query languages. In: Proc. 14th ACM Symp. on Theory of Computing, pp. 137–146 (1982)

65. Clarke, E., Grumberg, O.: Avoiding the state explosion problem in temporal logic model-checking algorithms. In: Proc. 16th ACM Symp. on Principles of Distributed Computing, pp. 294–303 (1987)
66. Browne, M., Clarke, E., Dill, D., Mishra, B.: Automatic verification of sequential circuits using temporal logic. IEEE Transactions on Computing C-35, 1035–1044 (1986)
67. Clarke, E., Mishra, B.: Hierarchical verification of asynchronous circuits using temporal logic. Theoretical Computer Science 38, 269–291 (1985)
68. Burch, J., Clarke, E., McMillan, K., Dill, D., Hwang, L.: Symbolic model checking: 10^{20} states and beyond. In: Proc. 5th IEEE Symp. on Logic in Computer Science, pp. 428–439 (1990)
69. Burch, J., Clarke, E., McMillan, K., Dill, D., Hwang, L.: Symbolic model checking: 10^{20} states and beyond. Information and Computation 98(2), 142–170 (1992)
70. McMillan, K.: Symbolic Model Checking. Kluwer Academic Publishers, Dordrecht (1993)
71. Clarke, E.: The birth of model checking. In: Ramanujam, R., Sarukkai, S. (eds.) ICLA 2009. LNCS (LNAI), vol. 5378. Springer, Heidelberg (2009)
72. Lichtenstein, O., Pnueli, A.: Checking that finite state concurrent programs satisfy their linear specification. In: Proc. 12th ACM Symp. on Principles of Programming Languages, pp. 97–107 (1985)
73. Vardi, M., Wolper, P.: Reasoning about infinite computations. Information and Computation 115(1), 1–37 (1994)
74. Wolper, P., Vardi, M., Sistla, A.: Reasoning about infinite computation paths. In: Proc. 24th IEEE Symp. on Foundations of Computer Science, pp. 185–194 (1983)
75. Vardi, M., Wolper, P.: An automata-theoretic approach to automatic program verification. In: Proc. 1st IEEE Symp. on Logic in Computer Science, pp. 332–344 (1986)
76. Kupferman, O., Vardi, M.: Weak alternating automata are not that weak. ACM Transactions on Computational Logic 2(2), 408–429 (2001)
77. Sistla, A., Vardi, M., Wolper, P.: The complementation problem for Büchi automata with applications to temporal logic. Theoretical Computer Science 49, 217–237 (1987)
78. Courcoubetis, C., Vardi, M., Wolper, P., Yannakakis, M.: Memory efficient algorithms for the verification of temporal properties. In: Clarke, E., Kurshan, R.P. (eds.) CAV 1990. LNCS, vol. 531, pp. 233–242. Springer, Heidelberg (1990)
79. Courcoubetis, C., Vardi, M., Wolper, P., Yannakakis, M.: Memory efficient algorithms for the verification of temporal properties. Formal Methods in System Design 1, 275–288 (1992)
80. Bryant, R.: Graph-based algorithms for Boolean-function manipulation. IEEE Transactions on Computing C-35(8), 677–691 (1986)
81. Emerson, E., Lei, C.L.: Efficient model checking in fragments of the propositional μ-calculus. In: Proc. 1st IEEE Symp. on Logic in Computer Science, pp. 267–278 (1986)
82. Clarke, E., Grumberg, O., Hamaguchi, K.: Another look at LTL model checking. In: Proc 6th Int. Conf. on Computer Aided Verification. LNCS, pp. 415–427. Springer, Heidelberg (1994)
83. Gerth, R., Peled, D., Vardi, M., Wolper, P.: Simple on-the-fly automatic verification of linear temporal logic. In: Dembiski, P., Sredniawa, M. (eds.) Protocol Specification, Testing, and Verification, pp. 3–18. Chapman & Hall, Boca Raton (1995)

84. Holzmann, G.: The model checker SPIN. IEEE Transactions on Software Engineering 23(5), 279–295 (1997)
85. Vardi, M.: Automata-theoretic model checking revisited. In: Cook, B., Podelski, A. (eds.) VMCAI 2007. LNCS, vol. 4349, pp. 137–150. Springer, Heidelberg (2007)
86. Brayton, R., Hachtel, G., Sangiovanni-Vincentelli, A., Somenzi, F., Aziz, A., Cheng, S.T., Edwards, S., Khatri, S., Kukimoto, T., Pardo, A., Qadeer, S., Ranjan, R., Sarwary, S., Shiple, T., Swamy, G., Villa, T.: VIS: a system for verification and synthesis. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, pp. 428–432. Springer, Heidelberg (1996)
87. Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: Nusmv 2: An opensource tool for symbolic model checking. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 359–364. Springer, Heidelberg (2002)
88. Duret-Lutz, A., Poitrenaud, D.: SPOT: An extensible model checking library using transition-based generalized büchi automata. In: Proc. 12th Int'l Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 76–83. IEEE Computer Society, Los Alamitos (2004)
89. Aggarwal, S., Kurshan, R.: Automated implementation from formal specification. In: Proc. 4th Int'l Workshop on Protocol Specification, Testing and Verification, pp. 127–136. North-Holland, Amsterdam (1984)
90. Aggarwal, S., Kurshan, R., Sharma, D.: A language for the specification and analysis of protocols. In: Proc. 3rd Int'l. Workshop on Protocol Specification, Testing, and Verification, pp. 35–50. North-Holland, Amsterdam (1983)
91. Kurshan, R.: Analysis of discrete event coordination. In: de Bakker, J., de Roever, W., Rozenberg, G. (eds.) REX 1989. LNCS, vol. 430, pp. 414–453. Springer, Heidelberg (1990)
92. Sabnani, K., Wolper, P., Lapone, A.: An algorithmic technique for protocol verification. In: Proc. Globecom 1985 (1985)
93. Kurshan, R.: Computer Aided Verification of Coordinating Processes. Princeton Univ. Press, Princeton (1994)
94. Hardin, R., Har'el, Z., Kurshan, R.: COSPAN. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, pp. 423–427. Springer, Heidelberg (1996)
95. Kurshan, R.: Verification technology transfer. In: Proc. 2006 Workshop on 25 Years of Model Checking. Lecture Notes in Conmputer Science. Springer, Heidelberg (2007)
96. Pnueli, A.: Linear and branching structures in the semantics and logics of reactive systems. In: Brauer, W. (ed.) ICALP 1985. LNCS, vol. 194, pp. 15–32. Springer, Heidelberg (1985)
97. Sistla, A., Vardi, M., Wolper, P.: The complementation problem for Büchi automata with applications to temporal logic. In: Brauer, W. (ed.) ICALP 1985. LNCS, vol. 194, pp. 465–474. Springer, Heidelberg (1985)
98. Banieqbal, B., Barringer, H.: Temporal logic with fixed points. In: Banieqbal, B., Pnueli, A., Barringer, H. (eds.) Temporal Logic in Specification. LNCS, vol. 398, pp. 62–74. Springer, Heidelberg (1987)
99. Vardi, M.: Unified verification theory. In: Banieqbal, B., Barringer, H., Pnueli, A. (eds.) Temporal Logic in Specification. LNCS, vol. 398, pp. 202–212. Springer, Heidelberg (1989)
100. Pratt, V.: Semantical considerations on Floyd-Hoare logic. In: Proc. 17th IEEE Symp. on Foundations of Computer Science, pp. 109–121 (1976)

101. Burstall, R.: Program proving as hand simulation with a little induction. In: Information Processing 1974, Stockholm, Sweden, International Federation for Information Processing, pp. 308–312. North-Holland, Amsterdam (1974)
102. Constable, R.: On the theory of programming logics. In: Proc. 9th ACM Symp. on Theory of Computing, pp. 269–285 (1977)
103. Engeler, E.: Algorithmic properties of structures. Math. Syst. Theory 1, 183–195 (1967)
104. Salwicki, A.: Algorithmic logic: a tool for investigations of programs. In: Butts, R., Hintikka, J. (eds.) Logic Foundations of Mathematics and Computability Theory, pp. 281–295. Reidel (1977)
105. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
106. Apt, K., Olderog, E.: Verification of Sequential and Concurrent Programs. Springer, Heidelberg (2006)
107. Fischer, M., Ladner, R.: Propositional modal logic of programs (extended abstract). In: Proc. 9th ACM Symp. on Theory of Computing, pp. 286–294 (1977)
108. Fischer, M., Ladner, R.: Propositional dynamic logic of regular programs. Journal of Computer and Systems Science 18, 194–211 (1979)
109. Pratt, V.: A practical decision method for propositional dynamic logic: Preliminary report. In: Proc. 10th Annual ACM Symposium on Theory of Computing, pp. 326–337 (1978)
110. Harel, D., Sherman, R.: Looping vs. repeating in dynamic logic. Inf. Comput. 55(1–3), 175–192 (1982)
111. Streett, R.: A propositional dynamic logic for reasoning about program divergence. PhD thesis, M.Sc. Thesis, MIT (1980)
112. Street, R.: Propositional dynamic logic of looping and converse. In: Proc. 13th ACM Symp. on Theory of Computing, pp. 375–383 (1981)
113. Streett, R.: Propositional dynamic logic of looping and converse. Information and Control 54, 121–141 (1982)
114. Kozen, D.: Results on the propositional μ-calculus. In: Nielsen, M., Schmidt, E.M. (eds.) ICALP 1982. LNCS, vol. 140, pp. 348–359. Springer, Heidelberg (1982)
115. Kozen, D.: Results on the propositional μ-calculus. Theoretical Computer Science 27, 333–354 (1983)
116. Bradfield, J., Stirling, C.: PDL and modal μ-calculus. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic. Elsevier, Amsterdam (2006)
117. Emerson, E., Clarke, E.: Characterizing correctness properties of parallel programs using fixpoints. In: Proc. 7th Int. Colloq. on Automata, Languages, and Programming, pp. 169–181 (1980)
118. Park, D.: Finiteness is μ-ineffable. Theoretical Computer Science 3, 173–181 (1976)
119. Pratt, V.: A decidable μ-calculus: preliminary report. In: Proc. 22nd IEEE Symp. on Foundations of Computer Science, pp. 421–427 (1981)
120. Emerson, E., Halpern, J.: Decision procedures and expressiveness in the temporal logic of branching time. Journal of Computer and Systems Science 30, 1–24 (1985)
121. Emerson, E., Halpern, J.: "Sometimes" and "not never" revisited: On branching versus linear time. In: Proc. 10th ACM Symp. on Principles of Programming Languages, pp. 127–140 (1983)
122. Emerson, E., Halpern, J.: Sometimes and not never revisited: On branching versus linear time. Journal of the ACM 33(1), 151–178 (1986)
123. Harel, D., Kozen, D., Parikh, R.: Process logic: Expressiveness, decidability, completeness. J. Comput. Syst. Sci. 25(2), 144–170 (1982)

124. Vardi, M., Wolper, P.: Yet another process logic. In: Clarke, E., Kozen, D. (eds.) Logic of Programs 1983. LNCS, vol. 164, pp. 501–512. Springer, Heidelberg (1984)
125. Harel, D., Peleg, D.: Process logic with regular formulas. Theoreti. Comp. Sci. 38(2–3), 307–322 (1985)
126. Hafer, T., Thomas, W.: Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In: Ottmann, T. (ed.) ICALP 1987. LNCS, vol. 267, pp. 269–279. Springer, Heidelberg (1987)
127. Beer, I., Ben-David, S., Landver, A.: On-the-fly model checking of RCTL formulas. In: Y. Vardi, M. (ed.) CAV 1998. LNCS, vol. 1427, pp. 184–194. Springer, Heidelberg (1998)
128. Beer, I., Ben-David, S., Eisner, C., Fisman, D., Gringauze, A., Rodeh, Y.: The temporal logic Sugar. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 363–367. Springer, Heidelberg (2001)
129. Fix, L.: Fifteen years of formal property verification at Intel. In: Proc. 2006 Workshop on 25 Years of Model Checking. LNCS. Springer, Heidelberg (2007)
130. Fix, L., Kamhi, G.: Adaptive variable reordering for symbolic model checking. In: Proc. ACM/IEEE Int'l. Conf. on Computer Aided Design, pp. 359–365 (1998)
131. Copty, F., Fix, L., Fraer, R., Giunchiglia, E., Kamhi, G., Tacchella, A., Vardi, M.: Benefits of bounded model checking at an industrial setting. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 436–453. Springer, Heidelberg (2001)
132. Armoni, R., Fix, L., Flaisher, A., Gerth, R., Ginsburg, B., Kanza, T., Landver, A., Mador-Haim, S., Singerman, E., Tiemeyer, A., Vardi, M., Zbar, Y.: The ForSpec temporal logic: A new temporal property-specification logic. In: Katoen, J.-P., Stevens, P. (eds.) TACAS 2002. LNCS, vol. 2280, pp. 211–296. Springer, Heidelberg (2002)
133. Bustan, D., Flaisher, A., Grumberg, O., Kupferman, O., Vardi, M.: Regular vacuity. In: Borrione, D., Paul, W. (eds.) CHARME 2005. LNCS, vol. 3725, pp. 191–206. Springer, Heidelberg (2005)
134. van Berkel, C.H., Josephs, M.B., Nowick, S.M.: Applications of asynchronous circuits. Proceedings of the IEEE 87(2), 223–233 (1999)
135. A comparison of reset control methods: Application note 11 (summit Microelectronics, Inc.) (1999), `http://www.summitmicro.com/tech_support/notes/note11.htm`
136. Vardi, M.: Nontraditional applications of automata theory. In: Hagiya, M., Mitchell, J.C. (eds.) TACS 1994. LNCS, vol. 789, pp. 575–597. Springer, Heidelberg (1994)
137. Gastin, P., Oddoux, D.: Fast LTL to Büchi automata translation. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 53–65. Springer, Heidelberg (2001)
138. Morley, M.: Semantics of temporal *e*. In: Melham, T.F., Moller, F. (eds.) Banff 1999 Higher Order Workshop (Formal Methods in Computation), University of Glasgow, Department of Computing Science Technical Report (1999)
139. Eisner, C., Fisman, D.: A Practical Introduction to PSL. Springer, Heidelberg (2006)
140. Armoni, R., Bustan, D., Kupferman, O., Vardi, M.: Resets vs. aborts in linear temporal logic. In: Garavel, H., Hatcliff, J. (eds.) TACAS 2003. LNCS, vol. 2619, pp. 65–80. Springer, Heidelberg (2003)

141. Eisner, C., Fisman, D., Havlicek, J., Lustig, Y., McIsaac, A., Van Campenhout, D.: Reasoning with temporal logic on truncated paths. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 27–39. Springer, Heidelberg (2003)
142. Eisner, C., Fisman, D., Havlicek, J., McIsaac, A., Van Campenhout, D.: The definition of a temporal clock operator. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 857–870. Springer, Heidelberg (2003)
143. Eisner, C., Fisman, D., Havlicek, J.: A topological characterization of weakness. In: Proc. 24th ACM Symp. on Principles of Distributed Computing, pp. 1–8 (2005)
144. Harel, D., Peleg, D.: More on looping vs. repeating in dynamic logic. Inf. Process. Lett. 20(2), 87–90 (1985)
145. Vijayaraghavan, S., Ramanathan, M.: A Practical Guide for SystemVerilog Assertions. Springer, Heidelberg (2005)

Game Quantification Patterns*

Dietmar Berwanger[1] and Sophie Pinchinat[2]

[1] RWTH Aachen, Germany
[2] IRISA Rennes, France

Abstract. We analyse two basic approaches of extending classical logics with quantifiers interpreted via games: Propositional Game Logic of Parikh and Alternating-Time Temporal Logic of Alur, Henzinger, and Kupferman. Although the two approaches are historically remote and they incorporate operationally orthogonal paradigms, we trace the formalisms back to common foundations and argue that they share remarkable similarities in terms of expressive power.

1 Introduction

The metaphor of games is at the basis of a rich and intuitive language for reasoning about interaction. Over the past three decades, substantial efforts have been made to integrate the elements of this language into logical formalisms (see [20] for a comprehensive survey).

We discuss two basic approaches towards formal reasoning about games: the Propositional Logic of Games introduced by Parikh [16] in 1983, which is the first formalism to incorporate games into a logic of computation, and the framework of Alternating-Time Temporal Logics of Alur, Henzinger, and Kupferman [2] introduced 15 years later, which is arguably the most influential game-based formalism in Computer-Science applications by today.

Both formalisms emerged from well-established logics for reasoning about the dynamics of computation. Parikh's Game Logic GL extends the Program Dynamic Logic (PDL) of Fischer and Ladner [9] by adding a dualisation operation that turns the description of a program into one of an interactive protocol.[1] The main representative of Alternating-Time Logics, ATL*, generalises the Computation-Tree Logic CTL* of Emerson and Halpern [8] to speak about the course of events in a multi-agent system. The two formalisms at the outset represent different specification paradigms: PDL captures an internal view on the execution of a program whereas CTL* reflects an external view on the dynamics of a computation. Accordingly, PDL quantifies over relations between program states, whereas CTL* quantifies over computation traces. Nevertheless, when viewed as extensions of the basic monomodal logic K with recursion mechanisms [19], the two formalisms turn out to have similar expressive power: they

* This research was supported by the Deutsche Forschungsgemeinschaft (DFG), the European COMBEST project, and the Indo-French research network TIMED-DISCOVERI.

[1] The term Game Logic with the abbreviation GL has also been used in [2] to denote a formalism that is unrelated to Parikh's Game Logic.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 116–130, 2009.

can both be embedded into the second alternation level and into the two-variable fragment of the μ-calculus; moreover, PDL with an additional loop-operator subsumes CTL*.

The casting of PDL and CTL* into logics of interaction occurs at two levels. At a local level, the basic modal quantifier which ranges over possible outcomes of a computation step is replaced in GL and ATL* with atomic-game operators associated to the outcome of an interactive event. However, atomic games a priori do not feature utilities; these arise only at a global level as winning conditions over plays, i.e., sequences of interactive events. To build rules for forming plays and winning conditions, GL and ATL* use logical constructs which largely preserve their meaning from the underlying logics of computation: Boolean and linear-time connectives, choice and iteration operators, and higher-order quantifiers over sequences of events.

On a first view, GL and ATL* may be seen as formalisms for reasoning about complex games composed from atomic ones. However, the analysis of rational behaviour in a game embedded within another game is notoriously difficult, if not hopeless. (Most of the questions raised 1971 in the seminal work of Howard on metagames [13] have remained unsolved so far.) In fact, the logics we consider do not pursue this aim; as the atomic games to which their semantics refer lack utilities, they are not games in the strict sense, but rather *game forms*, that is, descriptions of outcome functions. Essentially, both GL and ATL* lead a cut between two basic elements of game-oriented reasoning: the local *outcome* of an interactive event which is represented in the model, and the global *utility* drawn from a sequence of events which is determined by the formula. This separation between interdependent action and interactive decision-making reflects in the fundamental semantic constructs of the two formalisms. In Game Logic, the dualisation operation corresponds to a swap of capabilities, rather than utilities, between the players. In Alternating-Time Logics, atomic-game events are performed by (coalitions of) agents that are not equipped with subjective utility functions. The formal interpretation of these constructs sometimes contradicts the game-theoretic intuition delivered by the natural-language description of the logics. For a critical discussion on such aspects and recent approaches towards defining more natural semantics for ATL*, see [1] and [5].

Nevertheless, there is a sense in which GL and ATL* recover the proposition of compositional game-based reasoning: the *semantic games* of these logics do arise as compositions of atomic game forms via logical formulae. Semantic games, also called *model-checking* games, are zero-sum games associated to the question of whether a formula holds in a model or not [12]. Typically, there are two players, a Verifier who performs existential choices (e.g., decomposition of disjunctions, assignment of existentially quantified variables) and a Falsifier who performs universal choices (e.g., decomposition of conjunction, assignment of universally quantified variables); the Verifier can ensure to win if, and only if, the formula holds in the model. The correspondence between logics of computation and their semantic games is usually mediated via a specific automata model. For a general background on model-checking via games and automata see [11].

In this paper, we discuss terminological and technical challenges arising from the combination of interactive and compositional reasoning. We put forward the thesis that the formalisms of Game Logic and Alternating-Time Logics are effectively confined to the scope of determined two-player games with perfect information composed from atomic game forms. Game-theoretic concepts beyond this scope, e.g., those inherent to non-zero-sum games, imperfect information, or to games with more than two players essentially cannot be captured.[2] To substantiate this claim, we argue that the model-checking games for GL and ATL* —which characterise their semantics— are determined two-player games with perfect information.

The first part of the paper, Section 2, details the concept of an atomic game which is at the basis of the semantics of the two logics. We introduce distinct terms for notions that tend to be confounded in the literature. We maintain that a partial description of a game which lacks a utility function shall be called a game form. Likewise, an actor who can choose an action but who is not equipped with a utility function shall rather be called agent than player. We introduce the notion of an untyped game to denote abstract descriptions of interactive situation where actions are not yet assigned to the players. This representation subsumes the notions of effectivity function and that of concurrent game which underlie the semantics of Game Logic and Alterntating-Time Logics, respectively.

The definition of GL and ATL* is deliberately postponed until the terminological issues are settled. Originally, the semantics of the two logics is defined on different kinds of models: neighbourhood structures, or Montague-Scott models, and concurrent game structures. We introduce a common interpretation domain of *extensive game structures* based on untyped game forms, which generalise both neighbourhood and concurrent game structures. In Section 3, we present the semantics of GL and ATL*. To relate their expressive power, we show that ATL* is invariant under replacing atomic game forms while preserving the effectivity. As a consequence, it follows that the meaning of a ATL*-formula is determined by its meaning over neighbourhood models, that is, over the interpretation domain of GL-formulae. This invariance result relies on the notion of *sequentialisation* of an untyped game form which represents the only scenarios where choices are made and communicated to the other agents in a certain order.

Finally in Section 4, we introduce an automata-theoretic formalism that subsumes both GL and ATL* to describe how the recursion mechanisms of GL and

[2] This statement may seem to contradict the purpose of Alternating-Time Logics which is motivated as a formalism for speaking about (concurrent) games with several players. The contradiction can be traced back to a common terminological inaccuracy. A player incorporates different functions in a game: he is an *agent* with the capacity to perform actions, and, at the same time, he is a rational *decision maker* able to choose which action to perform. When we have game models of computational systems in mind, there are good reasons to distinguish between the two functions. The range of actions available to a player is typically determined by the design of the system whereas rational decisions on which actions to choose depend on the system specification. In light of this, the players invoked in the original definition of Alternating-Time Logics should be understood as non-deliberate agents [7].

ATL* are reflected through structural properties of automata, thus explaining limitations of their expressive power. The translation of GL and ATL* into automata, implicitly defines a notion of model-checking games for the two logics.

2 Atomic Games

As outlined in the introduction, at the core of our analysis are games that involve two players; we will call them *Ego* and *Alter*. The basic model is that of zero-sum games with two possible utility values -1 and 1, representing a win or a loss, respectively. Such a game is represented in *normal form* by a tuple (S^E, S^A, Z, π, u^E), where S^E and S^A are the sets of *strategies* or *actions* available to Ego and Alter, respectively, Z is the set of possible *outcomes* determined by the *play* function $\pi : S^A \times S^E \to Z$, and $u^E : Z \to \{-1, 1\}$ is a *utility* function associating to every outcome a winning or losing value for Ego. We sometimes write $s\frown t$ to denote $\pi(s,t)$.

We investigate different ways in which (descriptions of) games are composed out of (descriptions of) their parts. This section fixes our terminology for speaking about parts that are atomic in the sense that they involve only one round of interaction. The central notion is that of a game form, – a partial representation of a game which omits utilities. With effectivity functions and agent forms, we introduce two particular representations of game forms that will be used to define Game Logic and Alternating-Time Logics.

Game Forms, Types. At the most abstract level, an *untyped game form* is a tuple $\Gamma = (S, Z, \pi)$ specifying a set of strategies that is not associated to any particular player, a set of possible outcomes, and a (partial) play function $\pi : S \times S \to Z$. A game *type* α identifies a subset $S_\alpha \subseteq S$ of strategies in an untyped game form. The purpose of a game type, or simply type, is to designate the strategies available to the players for playing their part in a game. For each concrete modelling domain, a collection $\textsc{Act}$ of types is fixed beforehand. Types come in pairs: for every type $\alpha \in \textsc{Act}$ there is a *dual type* $-\alpha \in \textsc{Act}$, the dual of which is again α. The instantiation of an untyped game form Γ with a type α yields the (typed) game form $\Gamma_\alpha := (S^E_\alpha, S^A_{-\alpha}, Z, \pi)$. While the play function π might be only partially defined in Γ, we require it to be complete in every game form Γ_α with $\alpha \in \textsc{Act}$.

For instance, a matrix $p : [m] \times [n] \to Z$ can be viewed as an untyped game form $\Gamma = (S, Z, p)$ where the set of strategies consists of all row and column indices, $S = [\max\{m, n\}]$. (We denote by $[n]$ the set $[1, \dots, n]$.) There are two natural types $\textsc{Act} = \{\text{row}, \text{col}\}$ with $\text{col} = -\text{row}$, which associate the sets of rows and columns to the two players. The game form $\Gamma_{\text{row}} = ([m]^E, [n]^A, Z, p)$ represents the scenario in which Ego chooses a row and Alter, simultaneously, chooses a column whereas the dual $\Gamma_{\text{col}} = ([n]^E, [m]^A, Z, p)$ represents the scenario where Ego chooses a column and Alter chooses a row. By associating a utility $u : Z \to \{-1, 1\}$ for Ego to matrix entries, we obtain the games (Γ_{row}, u) and (Γ_{col}, u). Notice that these two games are in general different.

Our notion of game type is not standard in Game Theory; it is related to the established notion of a player type in games with incomplete information only in the loose sense that it makes an abstract description of a game more concrete. Intuitively, untyped game forms allow us to specify actions by abstracting from concrete players or agents who may perform them. Our example illustrates two uses of types, the motivations for which are bother rather particular to Computer-Science applications. On the one hand, different types can be applied to an untyped game form to define several games on the basis of a single description. On the other hand, types provide us with a way to separate the concept of a player from that of an agent: we may, e.g., first describe which actions are available to be performed by an agent and later use a type to specify whether it is Ego or Alter who can choose an action of this agent.

Agent Forms. We view descriptions of interactive events performed by several agents as a class of game forms with a particular representation. Let us fix a number n of agents. We refer to a list of elements $x = (x_i)_{i\in[n]}$, one for each agent, as a *profile*. A *coalition* is a set of agents $C \subseteq [n]$; the *complementary* coalition is $-C := [n] \setminus C$. For a profile x and a coalition C, we write x_C to denote the list $(x_i)_{i\in C}$. Then, an *agent form* is a tuple $(S_1 \dots S_n, Z, \pi)$ where S_i is the set of actions available to agent i, Z is the set of possible outcomes, and $\pi : \times_{i=1}^n S_i \to Z$ is a partial play function. For each coalition $C \subseteq [n]$, we derive the set $S_C := \{\, s_C \mid s \in \times_{i=1}^n S_i \,\}$ of joint actions available to C. The agent form represents the untyped game form $\Gamma = (S, Z, \pi)$ over the set of strategies $S := \{\, S_C \mid C \subseteq [n] \,\}$. Types on the domain of n-agent forms correspond to coalitions of agents, hence, $\textsc{Act} = 2^{[n]}$. Every coalition $C \subseteq [n]$ induces a type that is associated to the set S_C; the dual type is associated to S_{-C}. Accordingly, the typed game form Γ_C describes the scenario in which player Ego acts *in the capacity* of the coalition C whereas Alter acts in the capacity of the complementary coalition. Thus, agent forms can be understood as a representation artifice to describe 2^n different game forms by one structure.

Effectivity Functions and Neighbourhood Forms. The concept of effectivity function introduced by Moulin and Peleg [15] describes the power that a player has to force the outcome of a game within a target set. We assume the perspective of Ego when we refer to the effectivity of a game form. The *effectivity* $f(\Gamma)$ of a game form $\Gamma = (S^E, S^A, Z, \pi)$ (for Ego) is defined by

$$f(\Gamma) := \{\, X \subseteq Z \mid (\exists s \in S^E)\, (\forall t \in S^A)\, s\hat{\ }t \in X \,\}.$$

Clearly, the effectivity of a game is upwards closed in the sense that, with every set $X \in f(\Gamma)$, the closure $\lceil X \rceil := \{\, Y \mid X \subseteq Y \subseteq Z \,\}$ is included in $f(\Gamma)$.

When an untyped game form Γ is fixed, we write $f(\alpha)$ to denote the effectivity of Γ_α. Consider for example the game form described by the matrix in the left of Figure 1. For the two types selecting rows and columns, respectively, we obtain $f(\text{row}) = \lceil \{\, \{p, q\}, \{p, r\} \,\} \rceil$ and $f(\text{col}) = \lceil \{\, \{p\}, \{q, r\} \,\} \rceil$. We may also view the matrix as a description of an agent form with, say, agent 1 in charge of selecting rows and agent 2 in charge of selecting columns. Then, there are four

different types, one for each coalition $C \subseteq \{1,2\}$. Besides $f(\{1\})$ and $f(\{2\})$ which coincide with $f(\text{row})$ and $f(\text{col})$, we obtain $f(\{1,2\}) = \lceil\{\{p\},\{q\},\{r\}\}\rceil$ and $f(\emptyset) = \{\{p,q,r\}\}$.

Effectivity functions correspond to a particularly simple kind of game forms which we call *neighbourhood forms.* For a set Z of outcomes, a neighbourhood form is given by a set $F \subseteq 2^Z$. It describes the sequential scenario where Ego first chooses a set $X \in F$, then Alter chooses an element $x \in X$ which then constitutes the outcome of the game. For a fixed set ACT of types, we define untyped neighbourhood forms as the disjoint union of the typed neighbourhood forms over all types in ACT.

3 Logics and Models

The game forms discussed in Section 2 are concerned with the immediate outcome of interactive events. There is little to say, in logical terms, about such events in isolation. The challenge is to describe the dynamics of systems driven by sequences of interactive decisions. We focus on discrete systems that switch between states via transitions arising from the interplay of two competing players. In this section, we introduce extensive game structures as a generic model of such systems. After briefly describing syntax and semantics of Game Logic and Alternating Time Logics, we proceed to comparing the two logics. The key step is to show that GL and ATL* are both invariant under an equivalence which relates game structures of the same effectivity.

Extensive Game Structures. Extensive game structures generalise Kripke structures by replacing the accessibility relation with transition relations associated to effectivity functions. (Our model is close to the one proposed in [10] for plain ATL.)

Let ACT be a set of atomic game types closed under dual and let PROP be a set of atomic propositions. An *extensive game structure* for ACT and PROP is a structure $\mathcal{G} = (V, \Gamma, (V_p)_{p\in\text{PROP}})$ where V is a set of *positions*, Γ is a function that associates to every position v an untyped game form $\Gamma(v)$ for the domain ACT with outcomes in V, and V_p designates those positions where p holds. We will usually consider rooted structures with a designated initial position. Intuitively, taking a transition of type $\alpha \in$ ACT in state v of $\mathcal{G}$ amounts to switching into the state resulting as an outcome of an (atomic) play between Ego and Alter in the typed game form $\Gamma(v)_\alpha$. By taking a sequence of such transitions, the players Ego and Alter form a path of infinite length to which we refer as a *global* play.

3.1 Parikh's Game Logic

The Propositional Logic of Games GL, introduced by Parikh in 1983 ([16,17]), was the first logical formalism dedicated to reasoning about games. It proposes a way of describing the dynamics of interaction in a way similar to the one in which PDL describes the dynamics of program execution.

The syntax of GL allows to compose interactive scenarios for two players. Starting from a set PROP of atomic propositions and a set ACT of atomic game types (or action names), the expressions of GL are of two sorts, formulae and game expressions. Formulae φ are constructed from PROP by Boolean operations and modalities $\langle\gamma\rangle\varphi$ associated to game expressions γ that are generated by the grammar: $\gamma := a \mid \varphi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^* \mid \gamma^{\mathrm{d}}$, for $a \in$ ACT.

Informally, game expressions specify a schedule for a game between the two players Ego and Alter. The sequential composition $\gamma_1;\gamma_2$ means: play γ_1 first, then γ_2. The nondeterministic choice operator $\gamma_1 \cup \gamma_2$ lets the player in turn decide which of γ_1 or γ_2 to play. The iteration operator γ^* allows to play γ repeatedly, for a finite number of times, whereby the player in turn can decide before each round whether a new round is to be played. Finally, the test operator (φ?) invokes a referee to verify whether φ holds; if so, the play just continues, otherwise it breaks and the player in turn loses. Within atomic game forms, the plays proceed sequentially: first, the player in turn chooses his part of the action, and then the other player responds with his part. At the beginning of a play, Ego is in turn to move.

The game-specific essence of Game Logic resides in the dualisation operator. Informally, this operator corresponds to a player-swapping rule which reverses the order of play and the set of strategies available to a player. At the atomic level, it thus corresponds to dualising the type of a game form.

The semantics of GL-expressions is defined on neighbourhood structures, i.e., extensive game structures where the game forms $\Gamma(v)$ are given by untyped neighbourhood forms. Statements about the models are constructed by associating these game expressions with modalities. A typical statement $\langle\gamma\rangle\varphi$ expresses that, at the current state, Ego has a strategy to play according to γ in such a way that either φ is true when the play ends, or Alter breaks a rule and loses. For a formal definition we refer the reader to [18].

3.2 Alternating-Time Logics

The framework of temporal logics, founded in the work of Pnueli and Manna [14] represents a way of adding recursion mechanisms to basic modal logic that is conceptually different from dynamic logics such as PDL and GL. While the latter assume an internal perspective, referring to the execution of a program or a protocol, temporal logics are geared towards analysing the behaviour of systems in the flow of time, referring to sequences of states in a run by isolating them from their originating context.

The formalisms of Alternating-Time Logics proposed by Alur, Henzinger, and Kupferman [2] adapts the temporal quantification pattern for the purpose of analysing interactive systems, typically multi-agent systems. The main representative of this logic ATL* is defined as an extension of branching-time logic CTL* by adding a game quantifier which allows to refer to a play formed by two strictly competing players in an underlying game structure.

The native models of Alternating-Time Logics are concurrent game structures, i.e., extensive game structures where the transitions are given by agent forms.

For a set of atomic propositions $\textsc{Prop}$ and a number n of agents, the formulae of ATL* are of two sorts, state formulae φ and path formulae η, generated by the following grammars:

$$\varphi := \bot \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\!\langle C \rangle\!\rangle \eta \quad \text{and} \quad \eta := \varphi \mid \eta \vee \eta \mid \neg\eta \mid \mathsf{X}\eta \mid \eta \mathsf{U} \eta$$

where $p \in \textsc{Prop}$, $C \subseteq [n]$.

Plain ATL is the fragment of ATL* obtained by restricting the application of the operator $\langle\!\langle C \rangle\!\rangle$ to path formulae of type $\mathsf{X}\eta$ and $\eta\mathsf{U}\eta$. While not very expressive, this fragment is relevant because it is computationally tractable.

The meaning of ATL*-formulae in a extensive game structure $\mathcal{G}$ is defined by mutual induction over path and state formulae. Path formulae are interpreted over traces of plays in $\mathcal{G}$ according to the rules for linear temporal logic LTL with the constructors $\mathsf{X}\eta$ and $\eta\mathsf{U}\eta$ corresponding to the LTL-operators next and until, respectively. The quantifier $\langle\!\langle C \rangle\!\rangle$ transforms any path formula η into a state formula $\langle\!\langle C \rangle\!\rangle\eta$ which holds at those positions v from which, player Ego acting in capacity of coalition C has a strategy to force an infinite play which satisfies η.

It is important to remark that strategies of Ego are functions that associate to every initial segment π of a play, an action in the game form of type C reached in the play. In the extensive game over $\mathcal{G}$ with η describing the winning outcomes, Ego can force a win if, and only if, he can force a win while playing such that in every atomic game form, he moves first and makes his choice visible to Alter.

3.3 Comparing GL and ATL*

A priori, GL and ATL* are interpreted on different kinds of extensive game structures. To relate the two logics, we need to establish a correspondence between concurrent game structures and neighbourhood structures that is compatible with the logic. Minimal requirements on such a model correspondence would be (1) to relate two formulae $\varphi \in$ ATL* and $\varphi' \in$ GL if, for all concurrent game structures $\mathcal{G}$ and all corresponding neighbourhood structures $\mathcal{G}'$, we have $\mathcal{G} \models \varphi$ if and only if $\widetilde{\mathcal{G}}' \models \varphi'$, and (2) to respect the Boolean and modal operators common to the two logics.

In the following, we characterise a much stronger model correspondence. Towards this, we introduce an equivalence between general extensive game structures under which both GL and ATL* are invariant, and we show that each class of equivalent extensive structures has a representative among neighbourhood structures.

The idea is to identify each concurrent game structure $\mathcal{G} = (V, \varGamma, (V_p)_{p \in Prop})$ for n agents with the neighbourhood structure $\widetilde{\mathcal{G}}$ obtained by replacing every (untyped atomic) game form $\varGamma(v)$ with the neighbourhood form corresponding to the effectivity of $\varGamma(v)$. We justify this identification by showing that ATL*-formulae cannot distinguish between $\mathcal{G}$ and $\widetilde{\mathcal{G}}$. This allows us to reduce the interpretation domain of ATL* without loss to neighbourhood domains —the interpretation domain of GL— over the set of types $\textsc{Act} = 2^{[n]}$ corresponding to

coalitions of agents. Over this restricted domain, we can compare the expressiveness of ATL* speaking about coalitions of n agents with the expressiveness of GL* speaking about a set of 2^n atomic game actions associated to agent coalitions.

The difficulty consists in defining the effectivity of an untyped game form (where it is not yet known which actions belong to a player) in such a way that the meaning of all its typed instantiations (where actions are readily assigned to players) are preserved. Our approach involves the notion of sequentialisation of a game form, which captures the situation where the players perform their choice in a given order.

Sequentialisation. Any game form Γ naturally gives rise to two *sequential* game forms Γ_E and Γ_A. The game form Γ_E correspond to the scenario where Ego chooses his action first, and then Alter chooses his action being informed about Ego's choice. Conversely, in Γ_A, Alter chooses first and then Ego follows. We are interested, more generally, in the set of all sequential scenarios that may arise from an untyped game form $\Gamma = (S, Z, \pi)$, where strategies are not yet associated to a particular player. To capture the flow of information from the (yet unknown) first to the second mover we extend the set of available strategies to include all perfect-information strategies over choices from S. Formally, we consider the untyped game form $\hat{\Gamma} = (\hat{S}, Z, \hat{\pi})$ with strategies $\hat{S} = S \cup S^S$, where S^S denotes all functions from S onto S. The play function $\hat{\pi}$ is derived from π by setting $\hat{\pi}(s,t)$ to $\pi(s, t(s))$ if $(s,t) \in S \times S^S$, or to $\pi(s(t), t)$ if $(s,t) \in S^S \times S$; otherwise the value is left undefined. Each type α for Γ induces two types for the new game form, $E\!:\!\alpha$ and $A\!:\!\alpha$, which correspond to the scenarios in which Ego or Alter moves first, respectively. Thus, the new types assign to Ego the strategy sets $\hat{S}_{E\,:\,\alpha} := S_\alpha$ and $\hat{S}_{A\,:\,\alpha} := (S_\alpha)^{S-\alpha}$, respectively. We will call these types *sequential* types, and refer to $\hat{\Gamma}_{E\,:\,\alpha}$ and $\hat{\Gamma}_{A\,:\,\alpha}$, simply denoted $\Gamma_{E\,:\,\alpha}$ and $\Gamma_{A\,:\,\alpha}$, as *sequentialisations* of Γ. Observe that the dual of a sequential type $E\!:\!\alpha$ is $A\!:\!-\alpha$ which swaps both the sets of available actions and the play order of Ego and Alter.

By definition, effectivity functions do not distinguish between a game form Γ_α and its sequentialisation $\Gamma_{E\,:\,\alpha}$ with Ego as first mover, that is, $f(E\!:\!\alpha) = f(\alpha)$. Moreover, the effectivity of sequential types exhibits the following duality.

Lemma 1. *For any game form Γ and every appropriate type α,*

$$\begin{aligned} f(-E:\alpha) = f(A:-\alpha) &= \{\, X \subseteq Z \mid (\forall t \in S^A_\alpha)(\exists s \in S^E_{-\alpha})\, s\hat{\ }t \in X \,\} \\ &= \{\, Z \setminus X \mid X \notin f(E:\alpha) \,\}. \end{aligned}$$

Consequently, the set of sequentialisations of an untyped game form is characterised by the effectivities of the scenarios where Ego moves first.

Neighbourhood Representation. In Section 2 we illustrated that effectivity functions correspond to (typed) sequential game forms. Conversely, Lemma 1 points out that sequential game forms can be represented by a set of effectivity functions. In the following, we introduce untyped game forms that embed such a set of effectivity functions into one representation.

For an untyped game form Γ over a set of types $\textsc{Act}$, let us consider the graph G_α representing the sequential form associated to the effectivity of Γ_α where we label all arcs emanating from the root by α and the remaining arcs by $-\alpha$. Now, we merge all the graphs G_α for $\alpha \in \textsc{Act}$, by joining their roots and the terminal nodes that correspond to the same outcome. The resulting graph can again be viewed as an untyped sequential game form $\widetilde{\Gamma}$, which we call the *neighbourhood representation* of Γ. The meaning of types for $\widetilde{\Gamma}$ is determined by the arc labels; every type α corresponds to the set of strategies that select an α-successor for each node. The play function $\widetilde{\pi}(s^E, s^A)$ for $\widetilde{\Gamma}_\alpha$ returns the terminal node reached by moving first to the α-successor selected by s^E and then to the $(-\alpha)$-successor selected by s^A. The construction is illustrated in Figure 1.

As the following lemma points out, the neighbourhood representation $\widetilde{\Gamma}$ preserves the effectivity of all types for the original game form Γ.

Lemma 2. *Let Γ be an untyped game form and let $\widetilde{\Gamma}$ be its neighbourhood representation. Then, $f(\Gamma_\alpha) = f(\widetilde{\Gamma}_\alpha)$ for all types α.*

$$\pi = \begin{pmatrix} p & q \\ p & r \end{pmatrix}$$

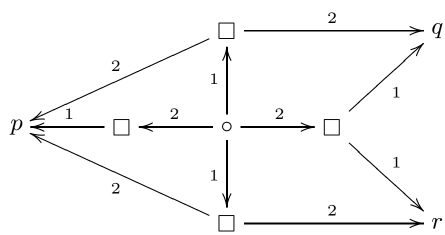


Fig. 1. A game form and its neighbourhood representation (only minimal effectivity sets are shown and the trivial types $\emptyset$ and $\{1, 2\}$ are omitted)

Effectivity Equivalence. Lemma 2 suggests a canonical representation of game forms in terms of neighbourhood forms. To make this idea precise, let us fix a modelling domain with a set $\textsc{Act}$ of types. We say that two games forms Γ and Γ' with the same set of outcomes are *effectivity-equivalent* if their effectivities $f(\Gamma)$ and $f(\Gamma')$ coincide. Likewise, two untyped game forms Γ and Γ' are effectivity equivalent if the game forms Γ_α and Γ'_α are so, for all types $\alpha \in \textsc{Act}$.

Due to the fact that effectivity functions preserve the duality of sequential types $A : \alpha$ and $E : -\alpha$, it follows that the effectivity equivalence between untyped game forms extends to their sequentialisations.

Lemma 3. *If two untyped game forms Γ and Γ' are effectivity equivalent, then so are their sequentialisations i.e., $f(\Gamma_{i\,:\,\alpha}) = f(\Gamma'_{i\,:\,\alpha})$, for every type $\alpha \in \textsc{Act}$ and each player $i \in \{E, A\}$.*

In particular it follows that, no matter whether a sequentialisation is applied to a game form or to its neighbourhood representation, the resulting sequential forms are equivalent.

Finally, we lift the notion of effectivity-equivalence to extensive game structures. We say that two extensive game structures $\mathcal{G} = (V, \Gamma, (V_p)_{p \in \textsc{Prop}})$ and

$\mathcal{G} = (V, \Gamma', (V_p)_{p\in\text{Prop}})$ over the same sets of positions V and with the same valuations V_p are effectivity-equivalent, if for any state v the game forms $\Gamma(v)$ and $\Gamma'(v)$ are effectivity equivalent. Notice that this is the same as requiring that $\mathcal{G}$ and $\mathcal{G}'$ have the same neighbourhood representation.

Theorem 4. *The logics* GL *and* ATL* *are invariant under effectivity equivalence: For any pair of effectivity-equivalent extensive game structures $\mathcal{G}$ and $\mathcal{G}'$, we have $\mathcal{G} \models \varphi$ iff $\mathcal{G}' \models \varphi$, for any formula φ of* GL *or* ATL*.

Proof. The proof is by induction over the structure of formulae. The critical case regards the modal next-step operators $\langle\!\langle C\rangle\!\rangle\mathsf{X}$ and γ of ATL* and GL, respectively. (When speaking about modal operators of ATL*, we tacitly mean the modal operators of the Alternating-Time μ-Calculus in which ATL* is embedded [2].)

Towards an operational characterisation of effectivity equivalence, we define *simulation* relations that capture the ability of a player to transfer his strategy from one game to another one in a way that maintains the same outcome on both sides. We say that, for Ego, the game form $\Gamma = (S^E, S^A, Z, \pi)$ is simulated by the game form $\Gamma' = (S'^E, S'^A, Z, \pi')$, and we write $\Gamma \preccurlyeq^E \Gamma'$, if for every $s \in S^E$ there exists $s' \in S'^E$ such that for every $t' \in S'^A$ there exists $t \in S^A$ for which $s\widehat{\ }t = s'\widehat{\ }t'$. We write $\Gamma \sim^E \Gamma'$, if $\Gamma \preccurlyeq^E \Gamma'$ and $\Gamma' \preccurlyeq^E \Gamma$. For Alter, the notions are defined analogously.

Then, for any pair Γ, Γ' of untyped game forms over the same set of types Act and with the same sets of outcomes, we have:

(i) For any type α, the forms Γ_α and Γ'_α are effectivity-equivalent if, and only if, $\Gamma_\alpha \sim^E \Gamma'_\alpha$.
(ii) As untyped game forms, Γ and Γ' are effectivity-equivalent if, and only if, $\Gamma_\alpha \sim^E \Gamma'_\alpha$ and $\Gamma_\alpha \sim^A \Gamma'_\alpha$, for all types α.

Accordingly, if two extensive game structures $\mathcal{G}$ and $\mathcal{G}'$ over the same set of positions V are effectivity-equivalent, the simulation relation between atomic games $\Gamma(v)$ and $\Gamma'(v)$, for all $v \in V$ extends naturally to a simulation relation between the structures. Essentially, every game composed via operators of ATL* or GL can be played on $\mathcal{G}$ in the same way as it can be played on $\mathcal{G}'$. □

Since every game structure is effectivity-equivalent to its neighbourhood representation, we obtain the following corollary.

Corollary 5. *A formula of* GL *or* ATL* *holds in an extensive game structure if, and only if, it holds in its neighbourhood representation.*

We can associate to any concurrent game structure $\mathcal{G}$ its neighbourhood representation $\widetilde{\mathcal{G}}$ to define an appropriate correspondence $\rightsquigarrow$ between formulae $\varphi \in$ ATL* and $\psi \in$ GL by setting $\varphi \rightsquigarrow \psi$ whenever $\mathcal{G} \models \varphi$ iff $\widetilde{\mathcal{G}} \models \psi$. Beyond respecting Boolean operations, this correspondence has the property that $\varphi \rightsquigarrow \psi$ and $\varphi' \rightsquigarrow \psi$ implies $\varphi \equiv \varphi'$. This allows us to extend the interpretation of Game Logic to concurrent game structures $\mathcal{G}$ by assigning to any formula $\varphi \in$ GL its meaning over the neighbourhood representation $\widetilde{\mathcal{G}}$ which will finally enable us to compare the two logics.

4 Recursion Mechanisms

In this last part of the paper, we sketch a direction for investigating the equivalence between formulae of GL and ATL* in terms of automata. In the previous section, we have seen that the basic modal operators of ATL and GL are essentially equivalent on extensive game structures with a set of types that are adequate for agent forms. To analyse the recursion mechanisms of GL and ATL* we now translate both logics into automata that run over extensive game structures. We call these *game automata*, because they operate with transitions determined by atomic game forms.

Game Automata. A game automaton for a set Prop of propositions and a set Act of types is a tuple

$$\mathcal{A} = (Q := Q^E \dot{\cup} Q^A, \textsc{Prop}, \textsc{Act}, q_\mathrm{I}, \delta, \Omega),$$

where Q is a finite state set with partitions Q^E and Q^A controlled by Ego and Alter, respectively, $q_\mathrm{I} \in Q$ is an initial state, $\delta : Q \times 2^{\textsc{Prop}} \to Q \times Q \cup \textsc{Act} \times Q$ is a transition function, and $\Omega : Q \to \mathbb{N}$ is a priority function describing a parity acceptance condition. Intuitively, the run of the automaton on an input structure $\mathcal{G}$ corresponds to a play of possibly infinite duration between Ego and Alter, starting from state q_I and the initial position v_0 of $\mathcal{G}$. From a state q and a position v, a transition $\delta(q, P)$ is enabled if the predicates in $P \subseteq \textsc{Prop}$ match those that hold at v; the player who controls the current state is in charge of the transition: if $\delta(q, P) = (q', q'')$, he has to choose between switching the automaton into state q' or into state q''; otherwise, if $\delta(q, P) = (\alpha, q')$, the player who controls q first performs an action of type α in the game form $\Gamma(v)$, and then the other player performs an action of the dual type $-\alpha$. The outcome of this local play determines the new position in $\mathcal{G}$, while the automaton is switched into q'. Finally, the game structure $\mathcal{G}$ is accepted, if Ego has a strategy to ensure that the sequence of states visited during the play satisfies the following parity property: the least priority occurring infinitely often is even.

Formally, acceptance is defined in terms of a graph game between Ego and Alter on the synchronised product between $\mathcal{A}$ and $\mathcal{G}$. The only non-standard element of this definition regards the intermediary configuration reached after an α-action has been executed by one player (and before the dual action is executed by the opponent) which does not correspond to any state-position pair. This intermediary state can be represented, by the set of all possible outcomes of the action (of type $-\alpha$) that the second mover has to take. The acceptance game is thus a classical graph game [11].

From Game Logic to Automata. To translate a GL-formula into a game automaton, we first first transform it into a pure game expression $\langle\gamma\rangle$true from which we also eliminate all non-atomic test operations; next, we put the game expression into a normal form in which every operation is associated explicitly

to a player $i \in \{\text{Ego}, \text{Alter}\}$ (see [3] for details). Notice that these operations only amount to relabellings on the syntax graph of the original formula which leaves its structure essentially unchanged.

Now, we build an automaton $\mathcal{A}(\gamma)$ inductively as illustrated in Figure 2; the states drawn in dotted frames are coalesced. Note that each component in this construction has a single entry (marked $\bullet$) and a single exit (marked $\circ$). Entry states are assigned to the player i in control of the corresponding subexpression. Significant priorities are assigned to states corresponding to $\star$-iteration operators. According to whether the iteration is controlled by Ego or Alter, the priority is even or odd, respectively, and the priority of a $\star$-expression is lower than that of all its subexpressions.

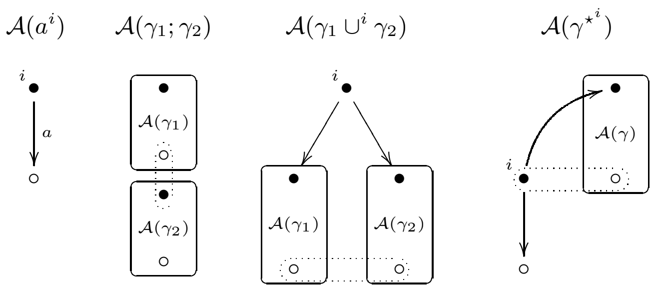


Fig. 2. Translating Game Logic into automata

The construction shows that GL-formulae translate into game automata where each component has a single entry and a single exit. In terms of programming-language theory, the interactive program constructions featured in GL are well structured. It is easy to show that every automaton with a transition graph that is well structured can be conversely translated into GL.

Proposition 6. *A class of extensive-form game models can be defined in Game Logic if, and only if, it can be described by a game automaton with a single-entry single-exit transition graph.*

To summarise, the higher-level quantification pattern of Game Logic corresponds to well-structured transition graphs. This structural restriction witnesses an expressive weakness of GL. It shows, for instance, that it is impossible to describe in GL extensive game models that embed a clique of size at least 3 ([4]). Thus, Game Logic is less expressive than the Alternating-Time μ-calculus.

From ATL to Automata. To translate a typical ATL*-formula $\langle\!\langle C \rangle\!\rangle \eta$ into a game automaton, we construct first the automata $\mathcal{A}_\varphi$ corresponding to the direct state subformulae φ of η. Next, we substitute all these state subformulae in η with fresh propositional variables X_φ as placeholders. The formula η' obtained in this way can be regarded as a linear-time expression over these variables; now,

we consider a deterministic word automaton $\mathcal{A}_\eta$ recognising the language of η'. This automaton we transform into a game automaton, by replacing each forward transition on the word model with an atomic modality corresponding to a game form of type C. Finally, we replace the tests for variables X_φ with a transition into the corresponding automaton $\mathcal{A}_\varphi$.

Analysing the structure of the automata obtained when translating plain ATL, the restricted variant of ATL where path formulae cannot be nested, it turns out that one obtains single-entry single-exit transition graphs. As a consequence of this translation and of Proposition 6, it thus follows that GL subsumes ATL.

Corollary 7. *Every formula φ of plain Alternating-Time Logic* ATL *can be translated into an equivalent Game Logic formula of size $\mathcal{O}(|\varphi|)$.*

The translation makes several expressive restrictions of ATL* apparent. For instance, every strongly connected component of the automaton obtained for an ATL* refers only to one kind of atomic types. Thus, one cannot express for instance, that agent 1 has a strategy to reach a state with property p in a play where he may form coalitions either with agent 2 or with agent 3, which is expressible in GL by $\langle((1,2)\cup(1,3))^*\rangle p$.

On the other hand, we conjecture that GL cannot express all properties expressible in ATL*. A promising source of inspiration towards settling this issue is the research on non-ambiguous regular expressions (see, e.g. [6]). Intuitively, a regular expression is non-ambiguous if every word can be matched in at most one way to expression symbols while it is read. An example of an inherently ambiguous property over the set set of predicates $\{0,1,2\}$ is that infinitely often the symbol 2 is seen 2 steps before the symbol 0 occurred. Whether Ego is able to enforce a path with this property seems unlikely to be expressible in GL, whereas it is clearly expressible in ATL*.

5 Conclusion

We set out to compare two prominent formalisms for reasoning about games that are historically remote and emerged from different operational paradigms. Parikh's Game Logic purports an internal perspective on the execution of an interactive program, whereas the family of Alternating Time Logic of Alur, Henzinger, and Kupferman reflect an external perspective on computations in a concurrent multi-agent systems.

By rephrasing the semantics of the two formalisms in unified framework, we point out that they show remarkable similarities: at the atomic level, the differences are limited to representation aspects, whereas at the global level, both formalisms have limitations due to recursion mechanisms which can be explained in terms of structural properties of game automata. Through our analysis, we reduce the question about how the two logics differ in their expressive power to questions about automata with a restricted transition structure.

References

1. Ågotnes, T., Goranko, V., Jamroga, W.: Alternating-time temporal logics with irrevocable strategies. In: TARK 2007: Proc. 11th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 15–24. ACM, New York (2007)
2. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM 49, 672–713 (2002)
3. Berwanger, D.: Game Logic is Strong Enough for Parity Games. Studia Logica 75, 205–219 (2003)
4. Berwanger, D., Grädel, E., Lenzi, G.: The variable hierarchy of the μ-calculus is strict. Theory of Computing Systems 40, 437–466 (2007)
5. Brihaye, T., Da Costa, A., Laroussinie, F., Markey, N.: ATL with strategy contexts and bounded memory, Research Report LSV-08-14, Laboratoire Spécification et Vérification, ENS Cachan (2008)
6. Bruggemann-klein, A., Wood, D.: One-unambiguous regular languages. Information and computation 142, 182–206 (1998)
7. Dennett, D.C.: The intensional stance. MIT Press, Cambridge (1989)
8. Emerson, E.A., Halpern, J.Y.: "Sometimes" and "not never" revisited: On branching versus linear time. In: Conference Record of the 10th Annual ACM Symposium on Principles of Programming Languages, POPL 1983, pp. 127–140. ACM, New York (1983)
9. Fischer, M., Ladner, R.: Propositional dynamic logic of regular programs. Journal of Computer and System Sciences 18, 194–211 (1979)
10. Goranko, V., van Drimmelen, G.: Complete axiomatization and decidability of alternating-time temporal logic. Theor. Comput. Sci. 353, 93–117 (2006)
11. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games. LNCS, vol. 2500. Springer, Heidelberg (2002)
12. Hintikka, J., Sandu, G.: Game-theoretical semantics. In: Handbook of Logic and Language, pp. 340–361. Elsevier and MIT Press (1997)
13. Howard, N.: Paradoxes of Rationality: Theory of Metagames and Political Behavior. MIT Press, Cambridge (1971)
14. Manna, Z., Pnueli, A.: Temporal Verification of Reactive Systems: Specification. Springer, Heidelberg (1992)
15. Moulin, H., Peleg, B.: Cores of effectivity functions and implementation theory. Journal of Mathematical Economics 10, 115–145 (1982)
16. Parikh, R.: Propositional Game Logic. In: IEEE Symposium on Foundations of Computer Science, pp. 195–200. IEEE, Los Alamitos (1983)
17. Parikh, R.: The logic of games and its applications. Annals of Discrete Mathematics 24, 111–140 (1985)
18. Pauly, M.: From Programs to Games: Invariance and Safety for Bisimulation. In: Clote, P.G., Schwichtenberg, H. (eds.) CSL 2000. LNCS, vol. 1862, pp. 496–586. Springer, Heidelberg (2000)
19. van Benthem, J.: Modal frame correspondences and fixed-points. Studia Logica 83, 133–155 (2006)
20. van der Hoek, W., Pauly, M.: Modal logic for games and information. In: Handbook of Modal Logic, pp. 1078–1143. Elsevier, Amsterdam (2007)

Extensive Questions
From Research Agendas to Interrogative Strategies

Emmanuel J. Genot

Université Charles-de-Gaulle Lille III – UMR 8163 STL

Abstract. Olsson and his collaborators have proposed an extension of Belief Revision Theory where an epistemic state is modeled as a triple $S = \langle \underline{\mathrm{K}}, E, \underline{\mathrm{A}} \rangle$, where $\underline{\mathrm{A}}$ is a *research agenda*, i.e. a set of research questions. Contraction and expansion apply to *states*, and affect the agenda. We propose an alternative characterization of the problem of *agenda updating*, where research questions are viewed as blueprints for research strategies. We offer a unified solution to this problem, and prove it equivalent to Olsson's own. We conclude arguing that: (*i*) our solution makes the idea of 'minimal change' in questions and agendas clearer; (*ii*) can be extended in ways the original theory was not, and may help better realize the aims this theory was proposed for; (*iii*) unveils some limitations of the initial approach, yet opening a way to overcome them.

1 Introduction: An Overview of Olsson's Theory

In [1] and [2], Erik Olsson and his collaborators propose to extend Belief Revision Theory (BRT) and model the epistemic state of an agent as a triple $S = \langle \underline{\mathrm{K}}_S, E_S, \underline{\mathrm{A}}_S \rangle$, rather than as a pair $\langle \underline{\mathrm{K}}_S, E_S \rangle$, where $\underline{\mathrm{K}}_S$ is a (closed) set of sentences (corpus), and E_S an entrenchment relation defined over $\underline{\mathrm{K}}_S$. The additional component $\underline{\mathrm{A}}_S$ is a *research agenda*, i.e. a set of questions, satisfying certain corpus-relative conditions, the agent would like to have answers to. This extension of the BRT framework, according to Olsson, could make BRT able to model some features of theory change which would extend the range of its intended applications.[1] Expansion (denoted '+') and contraction (denoted '$\div$') are taken to apply to *state* S, having thus an impact on the agenda. It is in particular assumed that the *Levi Identity* extends to states, i.e. that $S * a = (S \div \neg a) + a$, where '$*$' denotes revision.

A question $Q \in \underline{\mathrm{A}}_S$ is a set Q of sentences, named *potential answers* to Q, which are *jointly exhaustive*, *pairwise exclusive*, and *non-redundant* given S. These preconditions induce a partition of the maximally consistent expansions

[1] As an example, the acceptance of *ad hoc* hypotheses, or *ceteris paribus* laws, accompanied by a commitment to investigate exceptionless hypotheses, can be represented as expansion of the corpus on the one hand, and addition of a question on the agenda on the other hand. However, one of the initial motivations, i.e. to treat the problem of expanding into inconsistency, while inscribing on the agenda some question the answer to which would restore consistency, cannot be addressed (see n. 4).

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 131–145, 2009.

of $\underline{\mathrm{K}}_S$ (or simply S) by subsets of Q.[2] Formally: (i) $\veebar Q \in \mathrm{Cn}(\underline{\mathrm{K}})$ (where $\veebar Q$ denotes the *exclusive disjunction* of elements of Q); and: (ii) there is no $Q' \subset Q$ such that $\vee Q' \in \mathrm{Cn}(\underline{\mathrm{K}})$. Since $\underline{\mathrm{K}}$ is closed, $\underline{\mathrm{K}} = \mathrm{Cn}(\underline{\mathrm{K}})$ and we will use simply $\underline{\mathrm{K}}$. Let $\mathrm{Q}_{\underline{\mathrm{K}}}$ denote the set of questions satisfying the preconditions w.r.t. $\underline{\mathrm{K}}$. Olsson and Westlund refer to those questions as $\underline{\mathrm{K}}$-*questions*. If $Q \cap \underline{\mathrm{K}} \neq \varnothing$, then Q is $\underline{\mathrm{K}}$-*settled* (we will drop the prefix when no ambiguity ensues). It follows from those definitions that if $Q \in \mathrm{Q}_{\underline{\mathrm{K}}}$ and $Q \cap \underline{\mathrm{K}} \neq \varnothing$, then Q is a singleton.[3]

Let us introduce some further distinctions. A question which satisfies $\vee Q \in \underline{\mathrm{K}}$ and condition (ii) is $\underline{\mathrm{K}}$-*genuine*, while a question which which satisfies $\vee Q \in \underline{\mathrm{K}}$ and fails to satisfy (ii) will be said $\underline{\mathrm{K}}$-*rhetorical*. $\underline{\mathrm{K}}$-questions are, of course, special cases of $\underline{\mathrm{K}}$-genuine questions. Following Hintikka, we call $\vee Q$ the *presupposition* of Q.We also say that an answer to Q is *partial* if it makes Q rhetorical, and *complete* if it settles Q. Notice that, under this definition, complete answers are a special case of partial answers when some answers are incompatible.

At state S, there is no constraint for a question Q to be eligible to figure in $\underline{\mathrm{A}}_S$ other than satisfying the above preconditions, which translates in the first postulate offered to characterize agendas: the only qualification for being a $\underline{\mathrm{K}}$-agenda is that $\underline{\mathrm{A}}_S \subseteq \mathrm{Q}_{\underline{\mathrm{K}}_S}$, and we have:

$$\text{If } S = \langle \underline{\mathrm{K}}, E, \underline{\mathrm{A}} \rangle \text{ is an epistemic state, then } \underline{\mathrm{A}} \text{ is a } \underline{\mathrm{K}}\text{-agenda.} \tag{1}$$

However, the content of the agenda $\underline{\mathrm{A}}_{S \circ a}$ (where $\circ \in \{+, \div\}$) should be determined by $\underline{\mathrm{A}}_S$, and a. Contraction cannot solve questions, but can weaken preconditions (loosing exhaustiveness, exclusiveness, or both). Expansion can make some questions rhetorical.[4]

Observation 1. *Neither* $\mathrm{Q}_{\underline{\mathrm{K}}+a} \subseteq \mathrm{Q}_{\underline{\mathrm{K}}}$ *nor* $\mathrm{Q}_{\underline{\mathrm{K}}} \subseteq \mathrm{Q}_{\underline{\mathrm{K}}\div a}$ *hold in general.*

A consequence of Observation 1 is that it excludes two tentative continuity principles, namely:

$$\underline{\mathrm{A}}_{S+a} \subseteq \underline{\mathrm{A}}_S \tag{2a}$$

$$\underline{\mathrm{A}}_S \subseteq \underline{\mathrm{A}}_{S\div a} \tag{2b}$$

---

[2] Take $Q = \{a_1, \ldots, a_n\}$, and form the set $\underline{\mathrm{K}}^+Q$ of possible expansions of $\underline{\mathrm{K}}$ by *subsets of* Q. Given that if A is a finite set of sentences, $\underline{\mathrm{K}}+A = \mathrm{Cn}(\underline{\mathrm{K}} \cup A) = \mathrm{Cn}(\underline{\mathrm{K}} \cup \wedge A) = \underline{\mathrm{K}} + \wedge A$, $\underline{\mathrm{K}}^+Q$ is defined: $\underline{\mathrm{K}}^+Q = \{\underline{\mathrm{K}} + \wedge M : M \in \wp(Q)\}$. $\underline{\mathrm{K}} + \wedge M$ is a *maximally consistent expansion of* $\underline{\mathrm{K}}$ *by elements of* Q iff $M \in \underline{\mathrm{K}}^+Q$ and there is no $M' \in \underline{\mathrm{K}}^+Q$, $M \subset M'$ such that $\underline{\mathrm{K}} + \wedge M'$ is consistent (i.e. no M' such that $\underline{\mathrm{K}} \wedge M' \neq \underline{\mathrm{K}}_\perp$, where $\underline{\mathrm{K}}_\perp$ is the inconsistent belief set). Let $\underline{\mathrm{K}}^+_{\not\perp}Q$ denote the subset of $\underline{\mathrm{K}}^+Q$ of *maximally consistent expansions* of $\underline{\mathrm{K}}$ by subsets of Q. The kind of questions Olsson and his collaborators consider are those where: $\underline{\mathrm{K}}^+_{\not\perp}Q = \{\underline{\mathrm{K}} + a_i : a_i \in Q\}$.

[3] Olsson and Westlund remark that "[an] adequate model should keep track not only of questions in need of answers but also of beliefs that answer questions". Hence, it is important to record *singleton questions* on the agenda. Obviously, this allows also to define operations on questions to 're-open' potential answers.

[4] Notice that because of the definition of a $\underline{\mathrm{K}}$-agenda, there is only one possible agenda corresponding to the inconsistent belief set $\underline{\mathrm{K}}_\perp$, i.e. $\underline{\mathrm{A}}_\perp = \varnothing$. Hence, agenda is in no help in the problem of expansion into inconsistency, since according to (1), such an expansion erases all questions from the agenda.

Neither can be imposed as postulate for the agenda-part of, resp., state expansion and state contraction: (2a) would exclude from $\underline{\mathrm{A}}_{S+a}$ every question in $\underline{\mathrm{A}}_S$ made rhetorical (by expansion with a), be is settled or not; while (2b) would have the following effect: contraction by a (when $a \in \underline{\mathrm{K}}_S$) followed by expansion by a would *erase* some questions (those having lost their precondition following the contraction), which would not be recovered *even if* the contraction operator satisfies recovery.

In [1] and [2], Olsson *et al.* consider the task of preserving the continuity of agendas as a problem of *transformation of questions*, formally:

Problem 1 (Updating Questions I). Given $S = \langle \underline{\mathrm{K}}_S, E_S, \underline{\mathrm{A}}_S \rangle$ and $Q \in \underline{\mathrm{A}}_S$, and some a, specify two functions f^+ and $f^\div$, from $\mathrm{Q}_{\underline{\mathrm{K}}_S}$ to $\mathrm{Q}_{\underline{\mathrm{K}}_{S\circ a}}$ ($\circ \in \{+, \div\}$), such that if $Q \in \underline{\mathrm{A}}_S$, $f^\circ(Q) \in \underline{\mathrm{A}}_{S\circ a}$, preserving some continuity.

A construction for f^+ is given in [1], and solutions are proposed in [3] and [2] for $f^\div$. We propose an alternative formulation of the problem of updating agendas, using a simple game-theoretic setting. We will return to the solutions proposed by Olsson *et al.* to show their equivalence with our solution. The importance of this problem is that the possibility to offer postulates for *state expansion* and *state contraction* depends on it.

2 A Simple Game

Olsson and Enqvist insist that the problem of state contraction can be studied first applying contraction to a belief set, then identifying the effect on the agenda. We give in this section a game-theoretic formulation of this idea.

Consider the following two-player game: Given a *knowledge state* S, characterized by a *database* $\underline{\mathrm{K}}_S$, and a *research agenda* $\underline{\mathrm{A}}_S$ (in all respect analogous to elements of an epistemic state), we have two players, A and B. Player A manages $\underline{\mathrm{K}}_S$. It is assumed that $\underline{\mathrm{K}}_S$ is structured by an entrenchment relation.[5] Player B manages $\underline{\mathrm{A}}_S$. B has then to choose some method to obtain $\underline{\mathrm{A}}_{S\circ a}$ from $\underline{\mathrm{A}}_S$, depending on A's actions and altering $\underline{\mathrm{A}}_S$ as little as possible to obtain $\underline{\mathrm{A}}_{S\circ a}$. Since $\underline{\mathrm{K}}_S$ is logically closed, A's actions are simply responses to incoming information.[6] A's actions include adding to or erasing from $\underline{\mathrm{K}}_S$ some sentences, and other 'structural' changes, such as reorganizations of priority, rankings, entrenchments, etc., and AGM postulates are taken to embody 'rationality principles' for epistemic change, i.e. as *constraints* on admissible choices for A.[7]

[5] A has to choose some method to rearrange the entrenchment, but we leave this problem aside, since it does not affect agendas management.

[6] If $\underline{\mathrm{K}}_S$ is not closed, A could have other actions to perform, and $\underline{\mathrm{A}}_S$ would be determined on the basis of the surface information available from $\underline{\mathrm{K}}_S$ at a given time.

[7] This assumption is lifted in [4], where expansion and contraction are performed at the level of an interrogative game of the kind defined by Hintikka in [5], yielding representation results for the three operations in Hintikka's Extended Interrogative Logic.

On her turn, B will have to modify $\underline{\mathrm{A}}_S$. Assume furthermore that B knows what the incoming information is, and has witnessed A's actions. Now, consider that B's task, instead of altering questions, is to *select sequences of questions for 'interrogative games' (inquiries) possibly conduced by another player C*. Those games, at first, can be viewed as simple 'deduction games': inferential moves of C (from premises in $\underline{\mathrm{K}}_S$ and possibly answers to some questions) and *interrogative moves*, followed by 'reply' from a source S.[8] However, we assume that *deductive moves are preferred to interrogative moves whenever possible.* Notice that this assumption corresponds to the disallowance of rhetorical questions in agendas.

Intuitively, B's task can be characterized as follows. Associated to a set Q of *rival hypotheses* – which induces a partition of maximally consistent expansions of $\underline{\mathrm{K}}_S$ by subsets of Q, thus analogous to $\underline{\mathrm{K}}_S$-questions (see n. 2) – there is a *list of (sequences of) questions to be put to sources* (or an 'interrogative strategy'). Once the answers to those instrumental questions collected, one (and only one) of the hypotheses in Q will remain. Following the transition from $\underline{\mathrm{K}}_S$ to in $\underline{\mathrm{K}}_S \circ a$, B has to alter the list in such a way that the new strategy also *induces a partition of* $\underline{\mathrm{K}}_S \circ a$ (by combinations of answers to the questions in the list). The change in strategy will thus *determine the new set of rival hypotheses Q^**. In this setting, Problem 1 can be reformulated as follows:

Problem 2 (Updating Questions II). Given $S = \langle \underline{\mathrm{K}}_S, E_S, \underline{\mathrm{A}}_S \rangle$, let $\Sigma(\underline{\mathrm{K}}_S, \mathrm{Q}_{\underline{\mathrm{K}}_S})$ the set of possible strategies in interrogative games to answer $\underline{\mathrm{K}}_S$-questions. Specify two functions g^+ and $g^\div$ from $\Sigma(\underline{\mathrm{K}}, \mathrm{Q}_{\underline{\mathrm{K}}_S})$ to $\Sigma(\underline{\mathrm{K}}_{S\circ a}, \mathrm{Q}_{\underline{\mathrm{K}}_{S\circ a}})$ ($\circ \in \{+, \div\}$), so that if $Q \notin \underline{\mathrm{A}}_{S\circ a}$, $g^\circ(\sigma(\underline{\mathrm{K}}_S, Q))$ adds (or removes) any necessary (unnecessary) step from the interrogative strategy $\sigma(\underline{\mathrm{K}}_S, Q) \in \Sigma(\underline{\mathrm{K}}_S, \mathrm{Q}_{\underline{\mathrm{K}}_S})$ to obtain $\sigma(\underline{\mathrm{K}}_{S\circ a}, Q^*)$ where $Q^* \in \mathrm{Q}_{\underline{\mathrm{K}}\circ a}$ is *wholly determined by g°*.

Several examples of the kind of update performed by B will be given in the next section, in which we will also specify which idealizations are in force, and how they simplify the representation of B's task.

3 Updating Questioning Strategies

A simple representation of the kind of games B chooses strategy for is a tree (like the familiar semantic trees). Such a tree represents an ideal interrogative game in *extensive form*, i.e. where successive moves are explicitly displayed.[9] In

---

8 We will give a tree-form representation of these games in the next section, but will avoid any unnecessary technical discussion. These games are very close to those studied in [5]. For some applications in epistemology, see [6]. For an early proposal to combine BRT and interrogative games, see [3], which contains a slightly different treatment of the topics developed in the present paper.

9 By contrast, the *strategic* or *normal* form of the game displays one-shot choices of players, using a game-matrix. Extensive games are useful to represent a player's knowledge – several states (nodes) being indiscernible for a player in games with imperfect information – or to display asymmetric dependencies of moves (see [7], chap. 6). Imperfect information and sensitivity to order are ruled out by our idealizations, yet of capital interest for a more realistic modeling of inquiry.

such a game, C can play either deductive or interrogative moves. The aim of the game is to obtain an answer to some 'principal' question Q.

Let $\mathcal{G}(\underline{\mathrm{K}}, Q)$ denote the game 'about' Q given background $\underline{\mathrm{K}}$. The game is represented by a tree.[10] Each branch of the tree displays a *possible course* of the game (if the source answers the questions). A maximal branch (including the root and a leaf) is often referred to as a *maximal history* of the game.

At the root is $\underline{\mathrm{K}}$. B can add (or: plan for C) a *deductive move*, noted: 'C:!{a}' as soon as a follows from $\underline{\mathrm{K}}$ together with other formulae on the preceding nodes (interrogative moves are not formulae). Two branches diverge whenever a question is asked. An *interrogative move*, represented as: 'C:?{...}' is legal at a node as soon as the *presupposition* of the question follows from $\underline{\mathrm{K}}$ and the preceding nodes. A node displaying an interrogative move has as many successors as there are potential answers to the question asked. Since questions are assumed to have finitely many answers, the tree is finitely generated.[11]

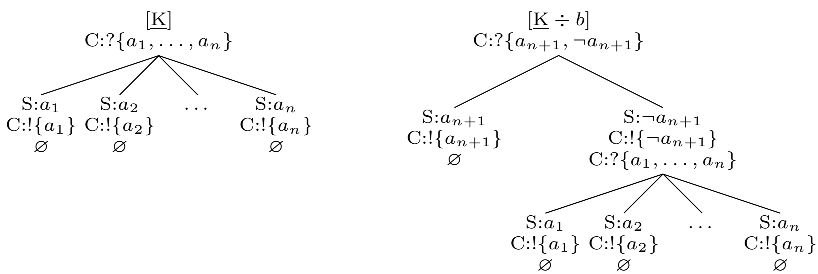


Fig. 1.

Update performed by B can be illustrated by abstract examples: Fig. 1 shows a 'strategy' with question $Q = \{a_1, \dots, a_n\}$, if $\vee Q \in \underline{\mathrm{K}}$ (left) and how the game can be updated if Q is weakened, when some additional (exclusive) answer a_{n+1} (right) is needed. Fig. 2 shows an update when some answers a_1 and a_2 become compatible. Notice that, in both cases, $\{a_1, \dots, a_n\}$ is still the content of an interrogative move, though it fails to satisfy the preconditions relative to the corpus which is at the root of the tree. Fig. 3 displays the effect of expansion – *partial* answer to Q (left) and *complete* (right). In contrast with the preceding examples, $\{a_1, \dots, a_n\}$ is not anymore the content of any interrogative move.

[10] A tree is a pair $\langle \mathcal{T}, < \rangle$, where $\mathcal{T}$ is a set of *nodes*, and $<$ a strict partial order; $a < b$ reads: "a is the predecessor of b" ("b is the successor of a"). a is the *immediate* predecessor of b iff $a < b$ and there is no a' such that $a < a' < b$. The *root* of the tree is the $a \in \mathcal{T}$ such that for all $a' \in \mathcal{T}$, $a' \neq a$, $a < a'$, and a' is a terminal node, or *leaf*, if there is no a'' s.t. $a < a''$. A branch β is any sequence $\langle a_i, \dots, a_m \rangle$ such that $a_i, \dots, a_m \in \mathcal{T}$ and the nth term of β is the *immediate predecessor* of the $n+1$th; β is *maximal* if there is no $a' \in \mathcal{T}$ s.t. $a' < a_i$ or $a_m < a'$.

[11] This assumption obviously cannot be made with first-order questions.

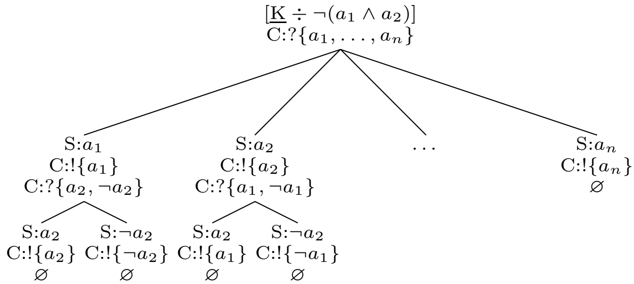


Fig. 2.

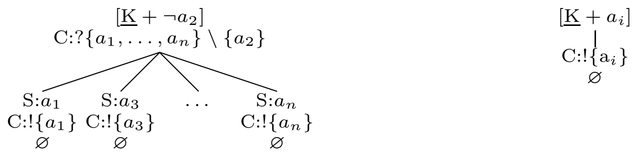


Fig. 3.

4 Yes-or-No Questions and Strategies

The strategies to be introduced make special use of *'yes-no'* questions, i.e. any two-element set $\{a, \neg a\}$. It is $\underline{\mathrm{K}}$-*open* if $\{a, \neg a\} \cap \underline{\mathrm{K}} = \varnothing$, and $\underline{\mathrm{K}}$-*settled* otherwise. Let $|^Y_N\text{-}\underline{\mathrm{K}}|$ be the set of $\underline{\mathrm{K}}$-open *'yes-no'* questions, and $|^Y_N\text{-}Q|$ denote the set of *'yes-no'* questions 'about' elements of Q.[12] For any question Q, we say that an answer $\pm a_i \in \{a_i, \neg a_i\}$ is *positive* if $\pm a_i \in Q \cap \{a_i, \neg a_i\}$, and *negative* otherwise.

Observation 2. *(1) For every* $\underline{\mathrm{K}}$ *and* a, $\{a, \neg a\}$ *is an open* $\underline{\mathrm{K}}$*-question or is* $\underline{\mathrm{K}}$*-settled. (2) If* Q *is a* $\underline{\mathrm{K}}$*-question and* $Q' \in |^Y_N\text{-}Q|$, *any* positive *answer to* Q', *following from* $\underline{\mathrm{K}} + b$, $\underline{\mathrm{K}} + b$*-settles* Q, *and any* negative *answer to* $Q' \in |^Y_N\text{-}Q|$ *makes* Q $\underline{\mathrm{K}} + b$*-rhetorical.*

(Notice that for strictly $\underline{\mathrm{K}}$-genuine questions a positive answer to some Q' from $\underline{\mathrm{K}} + b$ may not suffice to know all there is to know: another $Q'' \in |^Y_N\text{-}Q|$ may receive a positive answer, though $\underline{\mathrm{K}} + b$ does not entail it.) It follows immediately from Observation 2 (by contraposition of (2)) that:

$$\text{If } Q \in \mathrm{Q}_{\underline{\mathrm{K}}}, \text{ then } |^Y_N\text{-}Q| \subseteq \mathrm{Q}_{\underline{\mathrm{K}}} \tag{3}$$

Observation 2 is of special importance given the following result (reformulated for BRT):

[12] In practice, there is no reason to restrict analysis of a question Q to $|^Y_N\text{-}Q|$, since some potential answers to Q may be boolean compounds, which could be decomposed by asking more *'yes-no'* questions. But $|^Y_N\text{-}Q|$ will be sufficient to the present discussion. However, it will not allows to recover one of the solutions offered by Enqvist and Olsson to the problem of 'interrogative contraction' (see sec. 6).

Theorem 3 (Yes-No Theorem from [5, p. 55]). *For any corpus* $\underline{\mathrm{K}}$*, and any* $\underline{\mathrm{K}}$*-agenda* $\underline{\mathrm{A}}$*, if some conclusion follows from* $\underline{\mathrm{K}}$ *together with some answer to some* $Q \in \underline{\mathrm{A}}$*, then it follows from* $\underline{\mathrm{K}}$ *together with answers to* 'yes-no' *questions only.*[13]

Now, let's say that a strategy $\sigma(\underline{\mathrm{K}}, Q)$ (in a game $\mathcal{G}(\underline{\mathrm{K}}, Q)$) is *precondition free* iff it uses only $\varnothing$-questions (i.e. $\underline{\mathrm{K}}$-questions for $\underline{\mathrm{K}} = \varnothing$) as interrogative moves regardless of the preconditions of Q itself. Observation 2 and Theorem 3 yield:

Proposition 1. *For any question* Q*, there is a* precondition-free *interrogative strategy using only* 'yes-no' *questions for answering* Q*.*

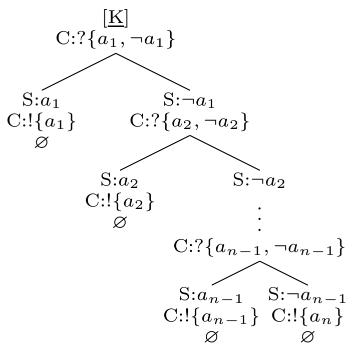


Fig. 4.

For $Q = \{a_1, \ldots, a_n\}$, Fig. 4 illustrates a precondition-free strategy for $Q \in \mathrm{Q}_{\underline{\mathrm{K}}}$ (the inferential steps to *partial* answers to Q are omitted). Notice that Q itself does not appear at any node in the tree, yet the strategy leads to have a complete answer to Q at each terminal node, and the game is indeed $\mathcal{G}(\underline{\mathrm{K}}, Q)$. Let us conclude with these useful properties relating *'yes-no'* questions and contraction:

$$|{}^{Y}_{N}\text{-}\underline{\mathrm{K}}| \subseteq |{}^{Y}_{N}\text{-}\underline{\mathrm{K}} \div b| \text{ for all } b \tag{4a}$$

$$\text{If } |{}^{Y}_{N}\text{-}Q| \subseteq \mathrm{Q}_{\underline{\mathrm{K}}}, \text{ then } |{}^{Y}_{N}\text{-}Q| \subseteq \mathrm{Q}_{\underline{\mathrm{K}} \div b} \text{ for all } b \tag{4b}$$

And their counterpart for expansion:

$$|{}^{Y}_{N}\text{-}\underline{\mathrm{K}} + b| \subseteq |{}^{Y}_{N}\text{-}\underline{\mathrm{K}}| \text{ for all } b \tag{5a}$$

$$\text{If } |{}^{Y}_{N}\text{-}Q| \subseteq \mathrm{Q}_{\underline{\mathrm{K}} + b}, \text{ then } |{}^{Y}_{N}\text{-}Q| \subseteq \mathrm{Q}_{\underline{\mathrm{K}}} \text{ for all } b \tag{5b}$$

All properties are straightforward consequences of postulates for expansion and contraction and Observation 2. Together with Proposition 1, they suggest the following:

[13] The theorem holding for belief sets and agendas is a consequence of the representation results proved in [4], once agendas are added.

Proposition 2. *Let Q be a $\underline{\mathrm{K}}$-question, and $\sigma(\underline{\mathrm{K}}, Q)$ be a* precondition-free *interrogative strategy in a game $\mathcal{G}(\underline{\mathrm{K}}, Q)$. Then for any b, if Q fails to be a $\underline{\mathrm{K}} \circ b$-question, there must be a solution to Problem 2 simply* removing from or adding to $\sigma(\underline{\mathrm{K}}, Q)$ *'yes-no'* questions.

5 A Systematic Method for Constructing Strategies

Establishing Propositon 2 takes three steps: the first step deals with loss of *exclusiveness*; the second, with loss of *exhaustiveness*; and the third with *partial* and *complete* answers. All three steps will use the same tree-construction method, which will be presented first. This construction is used to prove a general lemma about strategies. Given a question Q, we say that a strategy *exhausts* Q if collecting the answers to interrogative moves dictated by the strategy $\sigma(\underline{\mathrm{K}}, Q)$, the possible courses of the game (maximal branches) induce a partition (of maximally consistent expansions of $\underline{\mathrm{K}}$ with subsets of Q, with no unnecessary interrogative moves) equivalent to the partition generated by Q . The lemma itself states:

Lemma 1 (Exhausting Strategy). *For any question Q, there is a precondition-free exhausting strategy in $\mathcal{G}(\underline{\mathrm{K}}, Q)$.*

Notice that no restriction is put on Q (apart from its being a finite set): if Q is rhetorical or genuine, the *strategy* will take care of it, using deductive moves in the first case, or interrogative moves in the second. Hence Lemma 1 is not restricted to $\underline{\mathrm{K}}$-questions, hence its use for strategy updating.

Proof. Order $|{}^{Y}_{N}\text{-}Q|$ in any way, say $\{a_1, \neg a_1\}$,..., $\{a_n, \neg a_n\}$. Since these are all $\underline{\mathrm{K}}$-questions (by Observation 2), they can be asked any time at C's turn if they are open, or their answer can be inferred if they are settled. For some branch β, let $\beta^\wedge$ denote the conjunction of formulae in β.[14]

1. Write: 'C:?$\{a_1, \neg a_1\}$' as first move, then separate the initial branch in two sub-branches and add as first node of each, respectively, the two possible successors 'S:a_1' and 'S:$\neg a_1$'. This completes the first step.
2. Assume that the ith step has been carried out. Then check the leftmost branch β_n.
   (a) If $\{a_{i+1}, \neg a_{i+1}\} \cap \underline{\mathrm{K}} + \beta_n^\wedge = \varnothing$, write as the next node 'C:?$\{a_{i+1}, \neg a_{i+1}\}$', followed by a separation of β_n in two sub-branches β_1 and β_2, with and its two possible successors 'S:a_{i+1}' and 'S:$\neg a_{i+1}$' (resp.) as first node.
   (b) If $\{a_{i+1}, \neg a_{i+1}\} \cap \underline{\mathrm{K}} + \beta_n^\wedge \neq \varnothing$, write as next node 'C:!$\{a_{i+1}\}$' if $a_{i+1} \in \underline{\mathrm{K}} + \beta_n^\wedge$, and 'C:!$\{\neg a_{i+1}\}$' otherwise.
   (c) Go to the branch right to β, and perform the same test.

[14] Since the order of questions is not important as long as the strategy is precondition-free *and* the source is insensitive to it, we can treat branches as sets, rather than sequences, and expand with the conjunction of its member, defining formally $\beta^\wedge$ as follows: $\beta^\wedge = \{(a_i \wedge \cdots \wedge a_k) :$ C:!$\{a_j\}$ or S:a_j occurs in β $(i \leq j \leq k)$ and β is *maximal*$\}$.

Iterate (a)-(c) until reaching the rightmost branch.[15] This concludes the $i+1$th step.

3. Proceed until reaching $\{a_n, \neg a_n\}$.

The construction generates all combinations of elements of Q (and their negations) compatible with $\underline{\mathrm{K}}$. Moreover, *expanding with the conjunction of the 'S'-moves* of one maximal branch excludes the possibility of expanding with another maximal branch (since they can only differ from one another by the answer to a *'yes-no'* question). Hence it induces the desired partition. □

Let PFES abbreviate *precondition-free exhausting strategy*. Lemma 1 can be used to update a strategy $\sigma(\underline{\mathrm{K}}, Q)$ into a strategy for a game $\mathcal{G}(\underline{\mathrm{K}} \circ a, Q^*)$ when Q is not a $\underline{\mathrm{K}} \circ a$ question. The three following corollaries correspond to the three above-mentioned steps. The following is immediate, from Lemma 1.

Corollary 1. *If Q and $\underline{\mathrm{K}}$ are such that $\underline{\mathrm{K}}$ entails the presupposition of Q, and Q is not $\underline{\mathrm{K}}$-rhetorical, then there is a* PFES *in the interrogative game $\mathcal{G}(\underline{\mathrm{K}}, Q)$.*

Corollary 2. *If Q is such that $\underline{\mathrm{K}}$ does not entail the presupposition of Q, then, there is a set $\{b_1, \ldots, b_m\}$ such that there is a* PFES *in the game $\mathcal{G}(\underline{\mathrm{K}}, Q \cup \{b_1, \ldots, b_m\})$, extending any precondition free strategy in the game $\mathcal{G}(\underline{\mathrm{K}}, Q)$.*

Proof. Let $Q = \{a_1, \ldots, a_n\}$ and $\vee Q \notin \underline{\mathrm{K}}$. Hence C may receive 'S:$\neg a_i$' to any interrogative move 'C:?$\{a_i, \neg a_i\}$'. Hence, adding as terminal node of the rightmost branch: 'C:!$\{(\neg a_1 \wedge \ldots \wedge \neg a_n)\}$' after 'S:$\neg a_n$', yields a PFES in $\mathcal{G}(\underline{\mathrm{K}}, Q)$: in this case, $m = 1$ and $b_1 = (\neg a_1 \wedge \ldots \wedge \neg a_2)$. If some $\{b_1, \ldots, b_m\}$ is known such that $\vee Q \cup \{b_1, \ldots, b_m\} \in \underline{\mathrm{K}}$, apply the procedure of Lemma 1 to $Q \cup \{b_1, \ldots, b_m\}$.[16] □

Corollary 3. *If Q is any $\underline{\mathrm{K}}$-rhetorical question, there is a* PFES *(without unnecessary interrogative moves) in the game $\mathcal{G}(\underline{\mathrm{K}}, Q)$.*

Proof. Immediate from Lemma 1: any rhetorical question $\{a_i, \neg a_i\} \in |^{Y}_{N}\text{-}Q|$ will generate a deductive move: 'C:!$\{\pm a_i\}$', as a result of (2b). □

6 The Extensive Transformation Theorem

It is now possible to prove Propositon 2. We do it proving the following *Extensive Transformation Theorem*, which is equivalent to it, and uses the Lemma 1.[17] It

[15] The tree is finitely generated, since only interrogative moves generate branches. Using *'yes-no'* questions guarantees that the maximum number of branches at step k is 2^k. Since Q has only finitely many potential answer, this number remains finite. However, the complexity of the procedure makes it very clumsy, since each step requires an attempt to prove that $\pm a_j \in \underline{\mathrm{K}} + \beta_n^{\wedge}$.

[16] Obviously, if $\vee Q \cup \{b_1, \ldots, b_m\} \notin \underline{\mathrm{K}}$, then one can still add as terminal node of the rightmost branch: 'C:!$\{(\neg a_1 \wedge \cdots \wedge \neg b_m)\}$' after 'S:$\neg b_m$'.

[17] We formulate it with $\underline{\mathrm{K}}$-questions, though it is not necessary, in the light of the generality of Lemma 1, and the identity between the partition induced by exhausting strategies, and by $\underline{\mathrm{K}}$-questions.

is followed by three equivalence results, connecting the update method it gives rise to with former and parallel attempts by Olsson and his collaborators.

Theorem 4 (Extensive Transformation Theorem). *If Q is a $\underline{\mathrm{K}}$-question, but fails to be a $\underline{\mathrm{K}} \circ b$ question ($\circ \in \{+, \div\}$) for some b, then: (i) there is a* PFES *$\sigma(\underline{\mathrm{K}}, Q)$ in the game $\mathcal{G}(\underline{\mathrm{K}}, Q)$; and: (ii) there is a question Q^* and a* PFES *$\sigma(\underline{\mathrm{K}} \circ b, Q^*)$ in the game $\mathcal{G}(\underline{\mathrm{K}} \circ b, Q^*)$, where Q^* is a $\underline{\mathrm{K}} \circ b$-question, and can be obtained simply adding or deleting interrogative moves to $\sigma(\underline{\mathrm{K}}, Q)$.*

The proof below does not show how to obtain Q^*: this is left for the equivalence results. It is indeed necessary only if one is interested in solving Problem 1, since what is needed to solve Problem 2 is only to show that the new strategy is exhausting with respect to $\underline{\mathrm{K}} \circ b$, and (as already noticed) Q^* need not appear anywhere on the tree (since we work with precondition-free strategies).

Proof. (i) follows directly from Lemma 1. For (ii), there are two cases.

Case 1: $\underline{\mathrm{K}} \circ b = \underline{\mathrm{K}} \div b$. Under the conditions stated, Q must have lost one or both of its preconditions. Apply to $\sigma(\underline{\mathrm{K}}, Q)$, *first* (if needed) the procedure of Corollary. 2 to restore (the counterpart of) *exhaustiveness*; and *second* (if needed) the procedure described in Corollary. 1 how to restore (the counterpart of) *exclusiveness*. From these corollaries, it follows that the transformed strategy is a PFES in a game $\mathcal{G}(\underline{\mathrm{K}} \div b, Q^*)$ for some Q^*, as desired.
Case 2: $\underline{\mathrm{K}} \circ b = \underline{\mathrm{K}} + b$. Under the conditions stated, Q must have become $\underline{\mathrm{K}} + b$-rhetorical. Apply to $\sigma(\underline{\mathrm{K}}, Q)$, the procedure of Corollary. 3 to eliminate unnecessary interrogative moves. It follows immediately that the transformed strategy is a PFES in a game $\mathcal{G}(\underline{\mathrm{K}} \div b, Q^*)$ for some Q^*, as desired. □

The next three observations connect Theorem 4 with former attempts to solve the problem of question updating as expressed in Problem 1. Construction of an updating function following *expansion* was the first part of the problem. Corollary. 3 is the 'extensive' counterpart of this solution, based on the operation of *question truncation*. Following [1, p. 172] let $Q/_{\underline{\mathrm{K}}}a$, the $\underline{\mathrm{K}}$-*truncation of* Q *by* a, be defined: $Q/_{\underline{\mathrm{K}}}a = \{b \in Q : \neg b \notin \mathrm{Cn}(\underline{\mathrm{K}} \cup \{a\})\}$. (It is immediate that $Q/_{\underline{\mathrm{K}}}a \neq Q$ iff $\underline{\mathrm{K}} + a$ partially answers Q.) Truncation is instrumental to the definition of agenda updating upon expansion as shown in the following postulate for *State Expansion*:

$$S + a = \langle \underline{\mathrm{K}}_S + a, E', \underline{\mathrm{A}}' \rangle \text{ where } \underline{\mathrm{A}}' = \{Q' : Q' = Q/_{\underline{\mathrm{K}}_S}a \text{ for some } Q \in \underline{\mathrm{A}}_S\} \quad (6)$$

The equivalence of our updating method is given by the following observation:

Observation 5. *If Q is a $\underline{\mathrm{K}}$-question, but that for some $a_i \in Q$, $\neg a_i \in \underline{\mathrm{K}} + b$ for some b, one updates the* PFES *$\sigma(\underline{\mathrm{K}}, Q)$ using the method of Theorem 4, Case 2, then one obtains the same result as substituting $Q/_{\underline{\mathrm{K}}}b$ to Q in the agenda, and subsequently devising a* PFES *in the game $\mathcal{G}(\underline{\mathrm{K}} + b, Q/_{\underline{\mathrm{K}}}b)$. Moreover, $Q/_{\underline{\mathrm{K}}}b$ can be 'read off' the updated strategy.*

According to the setting described in sec. 2, B will update the strategy $\sigma\underline{\mathrm{K}}, Q$ in a way which is identical (up to a reordering of interrogative moves, as usual)

to substituting $Q/_{\underline{\mathrm{K}}_S} b$ to Q, and $\sigma(\underline{\mathrm{K}} + b, Q/_{\underline{\mathrm{K}}_S} b)$ to $\sigma\underline{\mathrm{K}}, Q$. Th proof of this observation shows what 'reading off' the question is: $Q/_{\underline{\mathrm{K}}} b$ can be constructed out of the tree, i.e one needs not determine the updated question before the update of the strategy. Simply put, a solution to Problem 1 can be obtained from solving Problem 2, and not the other way around. The same holds for the other observations.

Updating of question following contraction has been addressed in three ways, or in the terms of Olsson and Enqvist (see [2]), following three strategies. We present an equivalence with two of them. The third method will be briefly discussed at the end of this section. Olsson and Enqvist give the following principle, of *Agenda Preservation*, as a first postulate for constructing $\underline{\mathrm{A}}_{S \div b}$ out of $\underline{\mathrm{A}}_S$ and $\underline{\mathrm{K}}_{S \div b}$ (=$\underline{\mathrm{K}}_S \div b$):

$$\text{If } Q \in \underline{\mathrm{A}}_S \cap \mathrm{Q}_{\underline{\mathrm{K}}_S \div d} \text{ then } Q \in \underline{\mathrm{A}}_{S \div d} \tag{7}$$

Satisfying (7) poses no specific problem. The second postulate is as follows:

$$\begin{aligned} &\text{If } Q = \{a_1, \ldots, a_n\}, Q \in \underline{\mathrm{A}}_S \text{ and } Q \notin \mathrm{Q}_{S \div b} \text{ but for all } a_i, a_j \in Q, \\ &\qquad \neg(a_i \wedge a_j) \text{ then } Q \cup \{(\neg a_1 \wedge \cdots \wedge \neg a_n)\} \in \underline{\mathrm{A}}_{S \div b} \end{aligned} \tag{8}$$

Olsson and Enqvist call the resulting question an 'Ersatz question', and denote it $\mathbf{Simp}(Q)$. The following observation makes the connection between Theorem 4 and (8).

Observation 6. *If Q is a $\underline{\mathrm{K}}$-question, but has its exhaustiveness precondition of Q lost after contraction of $\underline{\mathrm{K}}$ by b, one updates the* PFES $\sigma(\underline{\mathrm{K}}, Q)$ *using the method of Theorem 4, Case 1, then if no more specific information is known, one obtains the same result as substituting* $\mathbf{Simp}(Q)$ *to Q in the agenda, and subsequently devising a* PFES *in the game $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Simp}(Q))$. Moreover,* $\mathbf{Simp}(Q)$ *can be 'read off' the updated strategy.*

The Second method, or *State Description Strategy*, treats both *exclusiveness* and *exhaustiveness* of a $\underline{\mathrm{K}}$-question Q using the notion of *state description of* Q. If $Q = \{a_1, \ldots, a_n\}$, then a state description of Q is any b such that: $d = \pm a_1 \wedge \cdots \wedge \pm a_n$. The third principle proposed is:

$$\begin{aligned} &\text{If } Q = \{a_1, \ldots, a_n\}, Q \in \underline{\mathrm{A}}_S \text{ and } Q \notin \mathrm{Q}_{S \div b} \text{ then} \\ &\{d : d = (\pm a_1 \wedge \cdots \wedge \pm a_n) \text{ and } (\underline{\mathrm{K}}_{S \div b}) + d \neq \underline{\mathrm{K}}_\perp\} \in \underline{\mathrm{A}}_{S \div b} \end{aligned} \tag{9}$$

Let $\mathbf{Sd}_{K \div b}(Q)$ denote the result of forming a question satisfying the condition in (9) (since $\underline{\mathrm{K}}_{S \div b} = \underline{\mathrm{K}}_S \div b$).

Observation 7. *If Q is a $\underline{\mathrm{K}}$-question, but not a $\underline{\mathrm{K}} \div b$-question, and if one updates the* PFES $\sigma(\underline{\mathrm{K}}, Q)$ *using the method of Theorem 4, Case 1, then one obtains the same result as substituting* $\mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q)$ *to Q in the agenda, and subsequently devising a* PFES *in the game $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q))$. Moreover,* $\mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q)$ *can be 'read off' the updated strategy.*

In the case of state description, specifically, our method is more economic than the method proposed by Olsson and Enqvist, in a perfectly natural sense: one

could devise a PFES for $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Sd}_{\underline{\mathrm{K}}\div b}(Q))$ using $|^{Y}_{N}\text{-}\mathbf{Sd}_{\underline{\mathrm{K}}\div b}(Q)|$, but this would lead to the construction of a 'new' strategy, since $|^{Y}_{N}\text{-}Q| \cap |^{Y}_{N}\text{-}\mathbf{Sd}_{\underline{\mathrm{K}}\div b}(Q)| = \varnothing$, or rather a strategy with different interrogative moves (the two strategies would anyway be equivalent, in that they would generate the same partition of maximally consistent expansions). On the other hand, our procedure preserves 'as much as possible' $\sigma(\underline{\mathrm{K}}, Q)$ in order to devise what is, in all effect, a PFES in $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Sd}_{\underline{\mathrm{K}}\div b}(Q))$.

Observation 5 shows that our method comply with Postulate (6). Observation 6 and 7 shows that it satisfies also (8) and (9). Solution to Problem 1 and 2 can now be unified, as well as the two 'postulates' (8) and (9): since g° can be used to generate an updated question $Q^* = f^{\circ}(Q)$, one can propose a tentative postulate for *State Contraction*:

$$S \div a = \langle \underline{\mathrm{K}}_S \div a, E', \underline{\mathrm{A}}' \rangle \quad \text{where } \underline{\mathrm{A}}' \subseteq \{Q' : |^{Y}_{N}\text{-}Q| \subseteq |^{Y}_{N}\text{-}Q'| \text{ for some } Q \in \underline{\mathrm{A}}_S\} \tag{10}$$

Moreover, since (as displayed in Fig. 2 and 1) use of PFES is not necessary, function g° can be defined (with some additional work) to apply to questions in their original form, without gong through the resource-consuming process of using exclusively *'yes-no'* questions.

7 Conclusion

The update method presented, though equivalent to those proposed by Olsson and his collaborators has some conceptual advantages. First, it presents a *unified construction method.* Rather than several operations, a unique tree-construction method is used, and can be used to reduce the different operations on questions as special cases. Second, the requirement of minimal change in enforced in an intuitive way: strategies are 'contracted' by expansion and 'expanded' by contraction, in such a way that each branch contains exactly those interrogative moves needed to induce a partition of the maximally consistent expansions of the corpus at the root with the set of (immediate) inferences from S's answers.

Finally, though not mandatory (see Fig. 1 and Fig. 2), the use of *precondition free* interrogative moves yields a continuous representation of updated questions: every interrogative move leaves a trace, since it is either maintained, or substituted with a deductive move, which permits re-insertion of the original interrogative move if the initial question has to be 'recovered'. In Olsson's original theory, only 'singleton' questions resulting from successive truncations remain on an agenda, while in our representation, for any question Q, every $Q \in |^{Y}_{N}\text{-}Q|$ leaves a trace update after update.

As a consequence, the framework we propose can be viewed as a generalization (or an extension) of Olsson's original one. Moreover, this extension offers better tools to apply the theory to its intended models. The original theory, though conceived of as a *dynamic* representation of an agent's research interest, the agenda $\underline{\mathrm{A}}_S$ 'tags along' with a corpus $\underline{\mathrm{K}}_S$, and the changes it undergoes can be

studied entirely without specifying parameters of 'field' inquiry. The 'extensive' approach combining BRT *cum* agendas with interrogative games offer the possibility to study the dynamics of inquiry.[18] It is however more closely related to the approach favoring belief *bases* (non-closed sets since Hansson's pioneering work in [8]), and handles changes through 'interrogative games' (combining deductive moves with interrogative moves, and means to retract some answers or premises at a given point, in such a way that it is possible to generate contractions and revisions).

Given the complexity of the agenda updating procedure,[19] and given the fact that changes in agendas are largely independent of the method used for contraction, it is unlikely that Agendas, in Interrogative BRT, may provide solutions to traditional problems in BRT, nor a less 'idealized' theory of belief states. Its interest lies rather in the way it allows to connect the formal theory to some issues in epistemology, and philosophy of science. However, an interrogative treatment of the *belief set change* is likely to offer some insights into these problems.[20] The real advance of Interrogative BRT may eventually prove not to be the additional structure it provides for belief states, but rather the novel perspective on belief sets and bases and operations performed on them, as being *question-driven*, and amenable to a treatment through some interrogative logic.

Acknowledgements. The author wish to express his gratitude to Sebastian Enqvist, Justine Jacot, Erik J. Olsson, Shahid Rahman and Gabriel Sandu for valuable comments and discussions on former versions of this paper. He has also greatly benefited from comments of three anonymous referees, whose suggestions lead to important additions, and hopefully some clarifications.

References

1. Olsson, E.J., Westlund, D.: On the Role of Research Agenda in Epistemic Change. Erkenntnis 65, 165–183 (2006)
2. Enqvist, S., Olsson, E.J.: Contraction in Interrogative Belief Revision (March 2008) (manuscript)
3. Genot, E.J.: The Best of All Possible Worlds: Where Interrogative Games Meet Research Agendas. In: Olsson, E.J. (ed.) Science in Flux. LEUS. Springer, Heidelberg (forthcoming)
4. Genot, E.J.: The Game of Inquiry. In: LOCCOL 2008 Conference, Prague, September 10–14 (2008)

[18] The framework developed in [4] is a basis for studying this dynamics (see n.7).

[19] The procedure requires a consistency test with the theory $\underline{\mathrm{K}}$ together with answers to preceding questions, for each answer to every question $\{a_i, \neg a_i\} \in |^{Y}_{N}\text{-}Q|$, when PFES are used, that is exponentially many such tests (on the length of Q). When *not* using PFES, the situation is hardly better.

[20] In [4], we investigate the failure of success of the *Recovery* postulate (showing at which conditions it can hold in *base*-contraction, where it typically fails), and offer some way to address the problem of *functional vs. relational* revision (hence the question of well-behaved entrenchments).

5. Hintikka, J., Halonen, I., Mutanen, A.: Interrogative Logic as a General Theory of Reasoning. In: Inquiry as Inquiry: A Logic of Scientific Discovery, pp. 47–90. Kluwer, Dordrecht (1999)
6. Hintikka, J.: Socratic Epistemology. Cambridge University Press, Cambridge (2007)
7. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
8. Hansson, S.O.: Belief Contraction without Recovery. Studia Logica 50, 251–260 (1991)

Appendix: Detailed Proofs

Auxliary Results

Proof (of Observation 2). (1) By classical logic, $(a \veebar \neg a) \in \underline{\mathrm{K}}$ whenever $\underline{\mathrm{K}}$ is consistent. (If $\underline{\mathrm{K}} = \underline{\mathrm{K}}_\perp = \mathcal{L}$, then $\{a, \neg a\} \cap \underline{\mathrm{K}} \neq \varnothing$ and $\{a, \neg a\}$ is trivially $\underline{\mathrm{K}}$-settled.) If $a \in \underline{\mathrm{K}}$, then $\{a, \neg a\}$ is not a $\underline{\mathrm{K}}$-question since $\{a\} \subset \{a, \neg a\}$, but then it is $\underline{\mathrm{K}}$-settled. The same holds with $\neg a$. (2) Let $Q = \{a_1, \ldots, a_n\}$, $Q \in \mathrm{Q}_K$, and $\pm a_i \in \underline{\mathrm{K}}{+}b$ for some $a_i \in Q$: (i) $\pm a_i = a_i$, $\underline{\mathrm{K}}{+}b{\cap}Q{=}\underline{\mathrm{K}}{+}b{\cap}\{a_i, \neg a_i\}{=}\{a_i\}$ iff the answer to $\{a_i, \neg a_i\}$ is *positive* and Q is *settled*; (ii) $\pm a_i = \neg a_i$, $\neg a_i \in \mathrm{Cn}(\underline{\mathrm{K}}{+}b)$ iff the answer to $\{a_i, \neg a_i\}$ is *negative* and (by Disjunctive Syllogism, and definition of a $\underline{\mathrm{K}} + b$-question) Q is $\underline{\mathrm{K}} + b$-rhetorical. □

Proof (of Proposition 1). For a given question Q, the set of *'yes-no'*-questions $|^{\scriptscriptstyle Y}_{\scriptscriptstyle N}\text{-}Q|$, if asked until a positive answer to one of them is obtained, will provide an answer to Q. By Observation 2, this strategy is precondition-free. □

Proof (of (4) *and* (5)*).* (4a): we prove the contrapositive. Assume that, for arbitrary $\underline{\mathrm{K}}$, b and a, $\{a, \neg a\} \cap \underline{\mathrm{K}} \div b \neq \varnothing$. If $a \in \underline{\mathrm{K}} \div a$, then $a \in \underline{\mathrm{K}}$, since by Inclusion postulate for Contraction, $\underline{\mathrm{K}} \div b \subseteq \underline{\mathrm{K}}$, and likewise if $\neg a \in \underline{\mathrm{K}} \div b$. Hence, if $\{a, \neg a\} \notin \mathrm{Q}_{\underline{\mathrm{K}} \div b}$, then $\{a, \neg a\} \notin Q_{\underline{\mathrm{K}}}$. Since $\underline{\mathrm{K}}$, $\{a, \neg a\}$ and b were arbitrary, it follows that it holds for arbitrary $\underline{\mathrm{K}} \div b$-settled *'yes-no'* question. Hence, by contraposition and Observation 2, $|^{\scriptscriptstyle Y}_{\scriptscriptstyle N}\text{-}\underline{\mathrm{K}}| \subseteq |^{\scriptscriptstyle Y}_{\scriptscriptstyle N}\text{-}\underline{\mathrm{K}} \div b|$ for all b, as desired. (5a): Similar to (4a), since $\underline{\mathrm{K}} \subseteq \underline{\mathrm{K}} + b$ (left to the reader). (4b) and (5b): from (4a) and (5a) respectively, and Observation 2, using (3). □

Equivalence Results

Proof (of Observation 5). Devising a PFES strategy for $Q/_{\underline{\mathrm{K}}_S} b$ amounts to apply the procedure of Lemma 1 using $|^{\scriptscriptstyle Y}_{\scriptscriptstyle N}\text{-}Q/_{\underline{\mathrm{K}}_S} b|$, that is $|^{\scriptscriptstyle Y}_{\scriptscriptstyle N}\text{-}Q| \cap \mathrm{Q}_{\underline{\mathrm{K}}+b}$ (this follows from (5a)). Hence, for any $a_i \in Q$, $\neg a_i \in \underline{\mathrm{K}}_S + b$, no interrogative move is used, but for any $a_j \in Q \cap Q/_{\underline{\mathrm{K}}_S} b$ (save for the last question in the ordering), the move 'C:?$\{a_j, \neg a_j\}$' is used.

On the other hand, applying the update procedure of Corollary. 3 as in the proof of Theorem 4, Case 2, the interrogative move 'C:?$\{a_i, \neg a_i\}$' will not appear in updated PFES for $\mathcal{G}(\underline{\mathrm{K}} + b, Q^*)$, being substituted with 'C:!$\{\neg a_i\}$'. Otherwise, the PFESinduced tree will include every interrogative move 'C:?$|^{\scriptscriptstyle Y}_{\scriptscriptstyle N}\text{-}a_j|$' for $a_j \in Q$, hence being identical with the above tree (up to a reordering of interrogative moves). □

Proof (of Observation 6). Devising a PFES for $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Simp}(Q))$ amounts to apply the procedure of Lemma 1 using $|^{Y}_{N}\text{-}\mathbf{Simp}(Q)|$, that is:

$$|^{Y}_{N}\text{-}Q| \cup \{\{(\neg a_1 \wedge \cdots \wedge \neg a_n), \neg(\neg a_1 \wedge \cdots \wedge \neg a_n)\}\}$$

This is identical to the application of Corollary. 2 with $m = 1$. On the other hand, Theorem 4 requires precisely the application of the same corollary to update $\sigma(\underline{\mathrm{K}}, Q)$ if Q looses some preconditions. Since *ex hypothesis* exhaustiveness is the only precondition lost, the result is identical (up to a reordering of interrogative moves). □

Proof (of Observation 7). Devising a PFES for the game $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q))$ can be done using $|^{Y}_{N}\text{-}\mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q)|$. But any PFES $\sigma(\underline{\mathrm{K}} \div b, \mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q))$ will do: it suffice that the tree induced by the strategy be such that: (i) interrogative moves are only *'yes-no'* questions; and (ii) for any maximal branch β, $(\underline{\mathrm{K}} \div b) + \beta^{\wedge}$ answers $\mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q)$. In particular, using the procedure of Theorem 4, Case 1, it is easily checked that:

$$\{\beta^{\wedge} : \beta \text{ is a } \textit{maximal branch} \text{ in } \sigma(\underline{\mathrm{K}} \div b, Q^*)\} = \mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q)$$

Hence, substituting the strategy constructed out of $\sigma(\underline{\mathrm{K}}, Q)$, one obtains a PFES for the game $\mathcal{G}(\underline{\mathrm{K}} \div b, \mathbf{Sd}_{\underline{\mathrm{K}} \div b}(Q))$. □

An Analytic Logic of Aggregation

Patrick Girard and Jeremy Seligman

Department of Philosophy, University of Auckland

Abstract. We present a modular approach to the logic of aggregated group preferences based on hybrid modal logic. The modularity of the system is twofold: 1) lifting preference relations between states to complex relations between propositions and 2) lifting individual preferences to group preferences. The preferences may be doxastic or proairetic, generating a logic of aggregated belief or aggregated desire, respectively, using a specific aggregation policy known as 'lexicographic re-ordering'. Each agent and each group of agents has an associated modal operator representing their preferences between states. The addition of the existential modality and nominals allows us to produce, first, a Hilbert-style axiomatization of the logic and then a more thorough analysis of inference using a Gentzen-style sequent calculus, in which the role of each operator is revealed.

Keywords: preference logic, lexicographic aggregation, hybrid modal logic, sequent calculus, analytic proof theory.

The paper is divided in four sections. After motivating lexicographic aggregation in Section 1, we give the basic language and axiomatization in Section 2. Section 3 summarises the background theory of the sequent calculus and its claim to provide an 'analytic' theory of proofs. This is then applied in Section 4 to a sequent calculus for the logic of aggregation.

1 Lexicographic Preference Aggregation

A common approach to analyzing desires and beliefs logically is by reducing them to preference and plausibility orders respectively (as in [4]) or a mixture of the two (as in [10]). In the case of desires, one starts with a preference order between objects, worlds, or more neutrally, *states*, and analyzes desires at the most preferable states: φ is desired if and only if it is true in all the most preferable states.[1] Likewise with beliefs based on plausibility orders, φ is said to be believed if it is true in the most plausible states.[2] Here we will use the term 'preference' in a way that is neutral between the proairetic (desirability) and

[1] The most preferable states are those compared to which no other state is preferred. The preference need not be total for there to be such states.

[2] This is the case, for instance, in doxastic dynamic logic based on Grove models, such as [12].

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 146–161, 2009.

doxastic (plausibility) reading, and represent it as a binary relation R between states that is required to be both reflexive and transitive (a *preorder*), with uRv interpreted to mean that v is no less preferable than u. It is also convenient to refer to the corresponding strict preorder defined as $uR^{<}v$ iff uRv and not vRu. These relations are describes using a standard modal language $\mathcal{L}_{\mathcal{P}}$ with two diamonds, $\Diamond$ and $\Diamond^{<}$, together with the existential modality E, which described the universal relation, and which plays an important role in lifting preference for states to preference for and between propositions.[3]

By associating diamonds $\langle a \rangle$ and $\langle a \rangle^{<}$ with each agent a, the approach migrates well to a multi-agent setting. Yet more is required to talk about preferences. For this we need to add modalities for groups of agents and analyze the relationship between the associated group preference relation and those of the group members. This is the problem of aggregation. Even with just two agents a and b, the language $\mathcal{L}_{\mathcal{P}}$ cannot describe those preferences that they share. For it to do so, there would have to be a modal operator O definable in $\mathcal{L}_{\mathcal{P}}$ such that the relation described by O is the intersection of the relations described by $\langle a \rangle$ and $\langle b \rangle$, and this cannot be done.[4] Our solution to this problem is quite simple: we add nominals, the naming devices of hybrid logic, using which the intersection of relations is known to be definable.[5] The addition of nominals therefore allows us to perform the second kind of lift, from individual to group preferences.[6]

In moving from individuals to groups one needs to follow a policy, known as an *aggregation procedure*. To motivate the specific aggregation policy we adopt, consider the following example. Three friends go to a restaurant and order a meal that they will then share. If the group does not contain deviant dietary agents, it may be sufficient for them to require unanimous agreement or perhaps a majority vote. But if one of the agents is vegetarian, these generally equitable procedures might not yield a desirable result: if the two meat-eaters go for their preferred feast of flesh, the vegetarian will starve. The obvious solution is to prioritize the preferences of the more vulnerable group member, the vegetarian, at least in vetoing meat and assuming that the carnivores have fall-back options in the realm of vegetables. Clearly no one prioritization will suit all situations, so we require a flexible mechanism for representing various ways of arranging group members according to priority.

This class of aggregation procedures, known as *lexicographic re-orderings*, were studied extensively in [1]. Given a particular hierarchy over a set of agents, aggregation is computed by giving priority to the agents further up the hierarchy in a compensating way: the group follows shared preferences, if it can, or follows the most influential agents, in case of disagreement. For example, suppose that

[3] Our approach to defining preference between propositions follows [18] and [17].
[4] A standard bisimulation argument proves this.
[5] See, for example, [2].
[6] A similar logic for preference aggregation is presented in [9]. The version contained in the present paper is a cleaner and a simplified version of the latter, with a new axiomatization. The Gentzen sequent calculus contained in the final section is an innovation of the present paper.

a has priority over b and c but not over d. To decide whether the group prefers state v to state u, we ask first whether a and d both take v to be no less preferable than u. If, in addition a has a strict preference for v over u then the group is taken to prefer v to u. If not, we check that b and c both take v to be no less preferable to u and if either b, c or d have a strict preference for v over u then the group does also. Lexicographic re-ordering (for any given way of prioritizing the members of the group) was shown by [1] (Section 3) to have the following attractive properties:

(I) Independence of irrelevant alternatives: adding a new state to those being compared does not change the group's preferences with respect to the old states.
(B) Based on preferences only: the group's preference relation is functionally determined by the preference relations of the group members.
(U) Unanimous with abstention: if a nonempty subset of the group's members are unanimous (i.e., they have identical preferences) regarding u and v and the remaining members are neutral, taking u and v to be equally preferable, then the group's preferences coincide with those of the unanimous subset.
(T) Preserves transitivity: the group's preference relation is guaranteed to be transitive if the individual members preference relations are transitive.[7]

Given the obvious existence of (many) lexicographic re-orderings, this establishes the consistency of these properties in contrast to the notorious impossibility theorem of Arrow in [3]. Moreover, every lexicographic re-ordering has the property of being 'non-dictorial', meaning that there is no agent whose preference ordering is guaranteed to be the same as the group's.[8] These are result of maximal generality because [1] also shows that *any* aggregation procedure that satisfies the IBUT conditions can be construed as a lexicographic re-ordering with respect to some way of prioritizing the members of the group.[9]

Also developed in [1] is the means of expressing any given aggregation procedure satisfying IBUT as a composition of two fundamental operations. The preferences of any two agents can be combined in a way that requires *agreement* or the *subordination* of one agent to the other. By composing these operations, it is possible to define any lexicographic re-ordering of a group's preferences,

[7] We only consider preference relations that are preorders (reflexive and transitive) so this condition merely requires that the group's preference relation is transitive. The reflexivity of the group's preference relation is derivable from the other conditions ([1], p. 23.)

[8] When a potential dictator abstains on the comparison between u and v and the rest of the group unanimously agree that v is better than u, the lexicographic aggregated preference will be a strict preference for v over u.

[9] Theorem 8 in [1].

and thereby any aggregation procedure satisfying IBUT.[10] We build a logic of aggregation based on these two operations and so construct a way of reasoning about group preference which does not fall pray to the overarching impossibility result of Arrow.[11]

2 Hybrid Modal Logic for Lexicographic Aggregation

We start with a system of terms for aggregated groups of agents, using the operations of agreement and subordination. The individual agents are denoted by symbols $a \in$ AGENT. Terms are then constructed using the operations $t_1 \parallel t_2$ for the group formed from t_1 and t_2 by a policy of agreement, and t_1/t_2 for the group formed by taking t_2 to be subordinate to t_1. Each group term t is associated with a modal preference operator $\langle t\rangle$ and its strict version $\langle t\rangle^<$. The language is then built using these operators from propositional variables $p \in$ PROP and nominals $i \in$ NOM, together with the standard Boolean operators, taking negation and disjunction as primitive. We also have the existential modal operator E. In other words, the hybrid modal language of lexicographic aggregation, $\mathcal{L}_{\mathcal{AG}}$, is defined by

$$\begin{aligned} \varphi &:= p \mid i \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle t\rangle\varphi \mid \langle t\rangle^<\varphi \mid E\varphi \\ t &:= a \mid t \parallel t \mid t/t \end{aligned}$$

A *frame* for $\mathcal{L}_{\mathcal{AG}}$ is a pair $\langle W, I\rangle$ consisting of a set W of states and a function I assigning a binary relations $I(t)$ and $I^<(t)$ on W to each term t. It is a *preference frame* if each of these relations is a preorder (reflexive and transitive) and $I^<(t)$ is the strict subrelation of $I(t)$, i.e., $\langle u, v\rangle \in I^<(t)$ iff $\langle u, v\rangle \in I(t)$ and $\langle v, u\rangle \notin I(t)$. It is an *aggregation frame* if, in addition, the following conditions are met:

$$\begin{aligned} I(t_1 \parallel t_2) &= I(t_1) \cap I(t_2) \\ I(t_1/t_2) &= (I(t_1) \cap I(t_2)) \cup (I(t_1)^<) \end{aligned}$$

Thus, in an aggregation frame, u is at least as good as v according to $t_1 \parallel t_2$ just in case t_1 and t_2 agree about this, and u is at least as good as v according to t_1/t_2 if either t_1 and t_2 agree or t_1 has a strict preference for v over u. A *model* $\mathfrak{M}$ is a triple $\langle W, I, V\rangle$ consisting of a frame $\langle W, I\rangle$ and a valuation function $V\colon \text{PROP} \cup \text{NOM} \to \mathcal{P}W$ such that $V(i)$ is a singleton for each $i \in$ NOM. $\mathfrak{M}$ is a *preference/aggregation model* if $\langle W, I\rangle$ is a preference/aggregation frame.

The semantics for $\mathcal{L}_{\mathcal{AG}}$ is given in Figure 1. As usual, we say that φ is valid in a model $\mathfrak{M}$ iff $\mathfrak{M}, u \models \varphi$ for all $u \in W$ and is valid in a frame iff it is valid in every model over that frame. The *logic of aggregation* is the set of all formulas φ in the language $\mathcal{L}_{\mathcal{AG}}$ that are valid in every aggregation frame.

[10] Corollary 14 in [1], in which the agreement operator is called ‘but’ and the subordination operator is called ‘on the other hand’.

[11] Of course, the price to pay for such a logic is that agents do not have equal votes, but as we saw in the example of the vegetarian at dinner, this is sometimes desirable.

Semantics of $\mathcal{L}_{\mathcal{AG}}$		
$\mathfrak{M}, u \models p$	iff	$u \in V(p)$
$\mathfrak{M}, u \models i$	iff	$\{u\} = V(i)$
$\mathfrak{M}, u \models \neg\varphi$	iff	$\mathfrak{M}, u \not\models \varphi$
$\mathfrak{M}, u \models \varphi \vee \psi$	iff	$\mathfrak{M}, u \models \varphi$ or $\mathfrak{M}, u \models \psi$
$\mathfrak{M}, u \models \langle t \rangle\varphi$	iff	$uI(t)v$ and $\mathfrak{M}, v \models \psi$ for some $v \in W$
$\mathfrak{M}, u \models \langle t \rangle^{<}\varphi$	iff	$uI^{<}(t)v$ and $\mathfrak{M}, v \models \psi$ for some $v \in W$
$\mathfrak{M}, u \models E\varphi$	iff	$\mathfrak{M}, v \models \psi$ for some $v \in W$

Fig. 1. Semantics for aggregation logic

$\mathcal{L}_{\mathcal{AG}}$ can define various notions of group preference, as shown in Figure 2.[12] Depending on the interpretation of 'preference' as proairetic or doxastic, these correspond to concepts of desire or belief. A group t's preference for state v over u is represented as $u <^t v$, meaning that the group desires v more than u (proairetic) or takes v to be more plausible than u (doxastic). Absolute preference for proposition φ is represented by $P^t\varphi$, meaning that the group desires φ to be the case (proairetic) or believes φ to be the case (doxastic), and this can be conditionalised as $P^t(\varphi|\psi)$. That proposition ψ is preferred to proposition φ, meaning that the group desires ψ to be the case more than φ (proairetic) or finds ψ to be more plausible than φ (doxastic), splits into a number of logically distinct concepts, two of which are represented here as $\varphi <^t_{\exists\exists} \psi$ and $\varphi <^t_{\forall\exists} \psi$.[13]

Group preference		
Preference between states	$u <^t v$	$E(u \wedge \langle t \rangle^{<} v)$
Preference for propositions	$P^t\varphi$	$U(\neg\langle t \rangle^{<}\top \to \varphi)$
Conditional preference for propositions	$P^t(\varphi\|\psi)$	$U((\psi \wedge \neg\langle t \rangle^{<}\psi) \to \varphi)$
Preference between propositions ($\exists\exists$)	$\varphi <^t_{\exists\exists} \psi$	$E(\varphi \wedge \langle t \rangle^{<}\psi)$
Preference between propositions ($\forall\exists$)	$\varphi <^t_{\forall\exists} \psi$	$U(\varphi \to \langle t \rangle^{<}\psi)$

Fig. 2. Defining concepts of preference in $\mathcal{L}_{\mathcal{AG}}$

The axiomatization of aggregation logic is given in Figure 3. We present the logic in building blocks. The first two tables give an axiomatization for the modal logic with two diamonds $\langle t \rangle\varphi$ and $\langle t \rangle^{<}\varphi$ and the existential modality E interpreted over the class of all frames. Along with the axioms of the third table, this is sufficient to axiomatize the hybrid logic $\mathcal{H}(E)$ of [15] (Table 5.3). The

[12] We make free use of the definability of other operators such as $\top$ (tautology), $\to$ (implication), $\wedge$ (conjunction), U (universal modality $\neg E \neg$), etc.

[13] The different combinations of quantifiers are discussed in [17]. Note that all of the listed concepts have weak versions, in which $\langle t \rangle^{<}$ is replaced by $\langle t \rangle$.

Propositional Logic	
1.	$\vdash \varphi \qquad (\varphi \text{ a tautology})$
2.	$\vdash \varphi \quad \& \quad \vdash \varphi \to \psi \quad \Rightarrow \quad \vdash \psi$
Modal Logic	
3.	$\vdash M(\varphi \vee \psi) \to (M\varphi \vee M\psi)$
4.	$\vdash (\varphi \vee EE\varphi \vee M\varphi) \to E\varphi$
5.	$\vdash E\neg E\varphi \to \neg\varphi$
6.	$\vdash \varphi \quad \Rightarrow \quad \vdash \neg M \neg \varphi$
Hybrid Logic	
7.	$\vdash i \to \varphi \quad \Rightarrow \quad \vdash \varphi \qquad (i \text{ not in } \varphi)$
8.	$\vdash E(i \wedge Mj) \to E(j \wedge \varphi) \quad \Rightarrow \quad \vdash E(i \wedge \neg M \neg \varphi) \qquad (i \neq j \text{ and } j \text{ not in } \varphi)$
9.	$\vdash Ei$
10.	$\vdash E(i \wedge \neg p) \to \neg E(i \wedge p)$
Preference axioms	
11.	$\vdash i \to Mi$
12.	$\vdash MMi \to Mi$
13.	$\vdash i \to (\langle t \rangle^{<} \varphi \leftrightarrow \langle t \rangle(\varphi \wedge \neg \langle t \rangle i))$
Aggregation axioms	
14.	$\vdash \langle t_1 \parallel t_2 \rangle i \leftrightarrow \langle t_1 \rangle i \wedge \langle t_2 \rangle i$
15.	$\vdash i \to (\langle t_1 / t_2 \rangle j \leftrightarrow ((\langle t_1 \rangle j \wedge \langle t_2 \rangle j) \vee \langle t_1 \rangle (j \wedge \neg \langle t_1 \rangle i)))$

Fig. 3. Axiomatization of aggregation logic in building blocks, with M ranging over each of the modal operators, E, $\langle t \rangle$ and $\langle t \rangle^{<}$ for each term t

fourth table adds axioms characterizing the class of preference frames: 11 and 12 for preorders and 13 for the relation between strict and weak preference. The last table adds the axioms that characterize the class of aggregation frames. To establish completeness for these various systems, we start with a known result ([15] Corollary 5.4.1):

Theorem 1. *The Axioms and rules 1-10 are sound and complete with respect to the hybrid logic $\mathcal{H}(E)$.*

The completeness of the axiomatizations of the classes of preference frames and aggregation frames follows directly from the fact their axioms are all 'pure formulas', which is to say that they contain no propositional variables, and that the axioms characterize the corresponding classes of frames, by another standard result in hybrid logic (e.g., Corollary 5.4.1 in [15]). Thus, it only remains to observe that the preference axioms are valid in a frame iff it is a preference frame and the aggregation axioms are valid in a preference frame iff it is an aggregation frame. This is all fairly trivial because the axioms directly mirror

the frame conditions. One point to observe is the role of nominals. It is only by using an antecedent nominal i and reference back to i later in Axioms 11, 13 and 15 that conditions such as reflexivity, strictness subordination can be captured. This cannot be done in ordinary modal logic. Hence we have:

Theorem 2. *(1) The axioms and rules 1-13 are sound and complete with respect to the class of preference frames, and (2) the axioms and rules 1-15 are sound and complete with respect to the class of aggregation frames.*

We therefore have an adequate logic of group preferences, at least from a theoretical perspective. For the remainder of the paper, we develop a more systematic and modular proof system using a Gentzen-style sequent calculus.

3 Analytic Proof Theory

Among its many virtues, Gentzen's sequent calculus provides a systematic analysis of the inferential role of logical operators.[14] If we define the *inferential meaning* of formula to be the set of valid argument forms that contain it as either a premise or conclusion, then the rules of Gentzen's calculus show how the inferential meaning of each formula is determined by the inferential meaning of its parts, and so conform to Frege's principle of the compositionality of meaning.[15] The inferential meaning of $(\varphi \vee \psi)$, for example, is determined entirely by the inferential meanings of φ and ψ, and Gentzen's rules show precisely how. In doing so, the definition of a valid argument form has to be stretched (in classical logic, at least) to allow for more than one conclusion. Gentzen's rules apply to *sequents*, which are expressions of the form $\Gamma \longrightarrow \Delta$, where Γ and Δ are finite sequences of formulas, usually written separated by commas. Such a sequent is *valid* if there is no appropriate interpretation in which the formulas in Γ, known as *premises*, are all true and the formulas in Δ, known as *conclusions*, are all false. The more usual definition of a valid argument is obtained as a special case, in which there is only one conclusion.

The calculus, displayed in Figure 4, consists of a complete set of rules for generating the valid sequents of first-order predicate logic, based only on the principle logical operators of the formulas contained in the sequent. Those rules that apply to conclusions are called *right* rules (marked R) and those that apply to premises are called *left* rules (marked L). In addition to the logical rules, there are two structural rules: the rule of Identity (I), which says that a sequent is valid if the same formula occurs as both premise and conclusion, and the rule of Sets (S), which says that validity of sequents is invariant with respect the equivalence

[14] [8] is the classical reference for Gentzen's calculus, [16] a text-book level introduction, and [13] develops the present method for devising a sequent calculus for modal and hybrid logics.

[15] Frege's principle, that the sense of an expression depends only on the sense of its parts (and their mode of composition) is usually applied within semantic theories of meaning, although some philosophers such as Dummett [6] and Brandom [5] have applied it to theories of meaning based on inferential role.

Structural Rules	
I	$\Rightarrow \quad \varphi, \Gamma \longrightarrow \Delta, \varphi.$
S	$\Gamma \longrightarrow \Delta \quad \Rightarrow \quad \Gamma' \longrightarrow \Delta'$ if $\Gamma \approx \Gamma'$ and $\Delta \approx \Delta'$.

Logical Rules	
$\neg$L	$\Gamma \longrightarrow \Delta, \varphi \quad \Rightarrow \quad \neg\varphi, \Gamma \longrightarrow \Delta.$
$\neg$R	$\varphi, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, \neg\varphi.$
$\vee$L	$\varphi, \Gamma \longrightarrow \Delta;\ \psi, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad (\varphi \vee \psi), \Gamma \longrightarrow \Delta$
$\vee$R	$\Gamma \longrightarrow \Delta, \varphi, \psi \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, (\varphi \vee \psi)$
$\exists$L	$\varphi[^x_a], \Gamma \longrightarrow \Delta \quad \Rightarrow \quad \exists x\, \varphi, \Gamma \longrightarrow \Delta$ if a does not occur in φ, Γ, Δ.
$\exists$R	$\Gamma \longrightarrow \Delta, \exists x\, \varphi, \varphi[^x_t] \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, \exists x\, \varphi$
$=$L$_1$	$s = t, \Gamma[^a_s] \longrightarrow \Delta[^a_s] \quad \Rightarrow \quad s = t, \Gamma[^a_t] \longrightarrow \Delta[^a_t].$
$=$L$_2$	$s = t, \Gamma[^a_t] \longrightarrow \Delta[^a_t] \quad \Rightarrow \quad s = t, \Gamma[^a_s] \longrightarrow \Delta[^a_s].$
$=$R	$\Rightarrow \quad \Gamma \longrightarrow \Delta, t = t$

Fig. 4. The sequent calculus G for predicate logic

relation $\approx$ which holds between sequences of formulas just in case they contain the same set of formulas. In other words, the order and number of copies of each formula are logically irrelevant.[16]

A proof in the calculus G is a deduction tree, of the familiar kind, with the sequent to be proved as root and with branches written

$$\frac{P_1 \quad \ldots \quad P_n}{C}\, R$$

for which C and $P_1, \ldots, P_n$ are sequents and $P_1, \ldots, P_n \quad \Rightarrow \quad C$ is a substitution instance of rule R. Leaves of the tree are formed by the unconditional rules I and $=$R. Although the justification of validity runs down the tree from leaves to root, we think of the tree as growing up from the root, with each rule applying to sequent C to yield branches $P_1, \ldots, P_n$. Each of the logical rules apply only to one formula in C, known as the *principal formula* of the rule, which may be on the left or right side. The remaining formulas, are known as the *context*. Most of the rules are *context-independent*, depending on and altering only the principal formula. There are two exceptions. The rule $\exists$L is *context-sensitive* because of the condition that the parameter does not occur in the context, and the rule $=$ L is *context-altering* because the context of the branches may be different from the context of the root. Aside from the context and the principal formula, all other formulas involved in a rule application are subformulas of the principal formula. This simple observation leads directly to an important result:

[16] Gentzen split the rule of Sets into the two rules of Permutation and Contraction, which capture invariance under the operations of permuting and deleting copies of premises (or conclusions). Subsequently there has been much interest in 'substructural' logics, in which some of the structural rules do not hold in full generality. (See, for example, [11].)

Subformula Property. Every formula occurring in a G proof is a subformula of a formula occurring at the root.[17]

Conspicuous by their absence are the structural rules of Weakening (W) and Cut (C):

$$\begin{array}{llll} \mathrm{W} & \Gamma \longrightarrow \Delta & \Rightarrow & \Gamma, \Gamma' \longrightarrow \Delta, \Delta' \\ \mathrm{C} & \Gamma \longrightarrow \Delta, \varphi \quad \varphi, \Gamma' \longrightarrow \Delta' & \Rightarrow & \Gamma, \Gamma' \longrightarrow \Delta, \Delta' \end{array}$$

Weakening allows redundant premises (or conclusions) in the root to be discarded from the branches and Cut allow us to use a formula that does not occur in the root as an intermediate stage, or lemma, in the argument. The left branch of an application of Cut proves the cut formula as a conclusion; the right branch uses it as a premise. These two rules are not included in the system G because they are *admissible*: adding them would not allow any more sequents to be proved. The proof of the admissibility of Weakening is straightforward: if π is a proof of $\Gamma \longrightarrow \Delta$ in G, then we can get a proof of $\Gamma, \Gamma' \longrightarrow \Delta, \Delta'$ by replacing each sequent $\Gamma'' \longrightarrow \Delta''$ that occurs in π by $\Gamma'', \Gamma' \longrightarrow \Delta'', \Delta'$ and checking that each branch is still an instance of the associated rule.[18]

The proof of the admissibility of Cut is Gentzen's famous Haupstatz, also known as 'Cut-elimination'.[19] The proof method depends on only two properties of the system, which will now be explained. Any application of Cut in a proof has the following form:

$$\dfrac{\dfrac{\begin{matrix}\pi_1 \\ P_1\end{matrix} \quad \cdots \quad \begin{matrix}\pi_n \\ P_n\end{matrix}}{\Gamma \longrightarrow \Delta, \varphi}\, R_1 \qquad \dfrac{\begin{matrix}\pi'_1 \\ P'_1\end{matrix} \quad \cdots \quad \begin{matrix}\pi'_m \\ P'_m\end{matrix}}{\varphi, \Gamma' \longrightarrow \Delta'}\, R_2}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\, \mathrm{C}$$

If, in addition, we assume that the proofs above $(\pi_1, \ldots, \pi_n, \pi'_1, \ldots, \pi'_m)$ are Cut-free, i.e., that they use only the rules of G then we can express the two crucial properties as follows:

Cut Reduction. If the cut formula φ is the principal formula of both R_1 and R_2 then the proof of sequent $\Gamma, \Gamma' \longrightarrow \Delta, \Delta'$ can be replaced by one whose only applications of Cut have cut formulas that are strictly less complex than φ.[20]

Permutation with Cut. If the cut formula φ is not the principal formula of R_1 then the order of application of R_1 and Cut can be swapped, so that there is a

[17] The definition of 'subformula' is the obvious one: φ is a subformula of $\neg\varphi$, both φ and ψ are subformulas of $(\varphi \vee \psi)$ and $\varphi[^x_t]$ is a subformula of $\exists x\, \varphi$ for each term t. Note that proofs in G require a countably infinite supply of new constant symbols, called *parameters*, which are included as terms, to serve in the application of $\exists$L.

[18] We also have to perform some substitutions of terms to deal with the quantifier rules, but these do not effect the structure of the proof.

[19] A full treatment of the basic hybrid logic we will be using is given in [13].

[20] The definition of the complexity of formulas is tailored to ensure that Cut reduction is possible. For predicate logic, the number of logical operators in the formula is a suitable measure of complexity.

proof of $\Gamma, \Gamma' \longrightarrow \Delta, \Delta'$ in which the application of Cut is closer to the leaves of the tree on the left side. Similarly, R_2 can be swapped with Cut if φ is not the principal formula of R_2.

Cut reduction and permutation with Cut allow applications of the Cut rule to be pushed up to the leaves of the proof tree and reduced in logical complexity until an application of the Identity rule on either the left or the right side is reached, at which point they can be removed completely.

The proof of Cut reduction relies on a balance between the left and right rules for each logical operator. The appropriate negation rules transformation, for example, is the following:

$$\dfrac{\dfrac{\overset{\pi_1}{\varphi, \Gamma \longrightarrow \Delta}}{\Gamma \longrightarrow \Delta, \neg\varphi}\,\neg\mathrm{R} \quad \dfrac{\overset{\pi_2}{\Gamma' \longrightarrow \Delta', \varphi}}{\neg\varphi, \Gamma' \longrightarrow \Delta'}\,\neg\mathrm{L}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\mathrm{C} \quad \rightsquigarrow \quad \dfrac{\dfrac{\overset{\pi_2}{\Gamma' \longrightarrow \Delta', \varphi} \quad \overset{\pi_1}{\varphi, \Gamma \longrightarrow \Delta}}{\Gamma', \Gamma \longrightarrow \Delta', \Delta}\,\mathrm{C}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\mathrm{S}$$

Note that the only application of Cut after the transformation has cut formula φ, which is less complex that the cut formula $\neg\varphi$ before the transformation. Similar transformations establish cut reduction for all pairs of rules. The proof of permutation with Cut is straightforward for context-independent rules and is effected by a simple swap. Negation, again, provides a simple example:

$$\dfrac{\dfrac{\overset{\pi_1}{\varphi, \Gamma \longrightarrow \Delta, \psi}}{\Gamma \longrightarrow \Delta, \neg\varphi, \psi}\,\neg\mathrm{R} \quad \overset{\pi_2}{\psi, \Gamma' \longrightarrow \Delta'}}{\Gamma, \Gamma' \longrightarrow \Delta, \neg\varphi, \Delta'}\,\mathrm{C} \quad \rightsquigarrow \quad \dfrac{\dfrac{\overset{\pi_1}{\varphi, \Gamma \longrightarrow \Delta, \psi} \quad \overset{\pi_2}{\psi, \Gamma' \longrightarrow \Delta'}}{\varphi, \Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\mathrm{C}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta', \neg\varphi}\,\neg\mathrm{R}$$

Note that the height of the right branch of the Cut after the transformation is one less than the height of the right branch of the Cut before the transformation, and the two left branches are unchanged. For context-sensitive and context-altering rules, a little more work is required, but the transformations involved are similar.

The admissibility of Cut has important consequences for the relationship between the sequent calculus and axiomatic systems. Suppose that we have an axiomatic system of deduction with a set A of logical axioms and a set R of inference rules, then for any set T of formulas, the set of theorems of T can be defined in the usual way. Then the axiomatic system can be simulated within the sequent calculus under the conditions of the following way:

Axioms to Rules Method. Given a set A of logical axioms, each of which is derivable in $G + \mathrm{CW}$, and a set R of inference rules, each of which is derivable in $G + \mathrm{CW}$, and a set T of formulas, if φ is a theorem of T then there is a finite subset Γ of T such that $\Gamma \longrightarrow \varphi$ is provable in G.[21]

[21] To say that an inference rule $\varphi_1, \ldots, \varphi_n$ is derivable in G means that if we add as new sequent rules the assertion of sequents $\longrightarrow \varphi_1, \ldots, \longrightarrow \varphi_{n-1}$ (at the leaves of a proof tree), then we can construct a proof of the sequent $\longrightarrow \varphi_n$. The derivability of axioms is a special case in which no additional sequent rules are required. We will give an example below.

The proof of this result is straightforward, given the admissibility of Cut. One merely reconstructs a deduction in the axiomatic system as a proof in G together with additional rules allowing the assertion of sequents $\longrightarrow \varphi$ (at the leaves of the tree) for each φ in T. Then, in any given proof of $\longrightarrow \psi$ for theorem ψ, take Γ to be the set of φ in T that occur in the proof, and add Γ to every sequent that occurs in the proof, so converting the assertions of $\longrightarrow \varphi$ at the leaves into sequents $\Gamma \longrightarrow \varphi$, which are applications of the Identity rule. The resulting proof is in $G + \mathrm{CW}$ and so can be converted into one in G by the elimination of Cut and Weakening. Typically, Cut is essential for showing the derivability of the rules of the axiomatic system.

The Axioms to Rules Method enables a completeness theorem for an axiomatic presentation of the logic to be transferred to the sequent calculus.[22] All that is required is to demonstrate that the rules of the sequent calculus have the two properties required for Cut-elimination, and that the axioms and rules of a complete axiomatic system are derivable. This can easily be done for G, using any one of a number of axiomatic systems for predicate logic.

Completeness of a sequent calculus provides more information than completeness of an axiomatic system because it extends to fragments. Given any restriction of the language of predicate logic to a subset of its logical symbols, the corresponding restriction of G is also complete, thanks to the Subformula Property. The decidability of some fragments, such as the quantifier-free fragment, is also a simple corollary of completeness and the observation that each formula in the fragment has only a finite number of subformulas.[23]

A final observation about the sequent calculus for predicate logic is needed before we consider modal logic and the logic of aggregation, in particular. The permutation of rules with Cut has already been explained but, in fact, the order of application of *any* two context-independent rules can be swapped if their principle formulas do not interact. The rules only effect their respective principal formulas and so their order of application can be reversed without difficulty. This ensure that these rules are *reversible*: any sequent that contains a formula φ (on the left or the right) has a proof in which the last rule applied is the corresponding left or right rule for the main connective of φ. The result extends to the remaining rules $\exists$L and $=$ L because their principal formulas are not removed when the rule is applied. Moreover, each of the rules has the property that the only logical operator mentioned in the rule occurs in the principal formula, so that the inferential role of the operator is completely captured by the rule. For example, the inferential meaning of a disjunction $(\varphi \vee \psi)$ is given by the set S of valid sequents of the form $(\varphi \vee \psi), \Gamma \longrightarrow \Delta$ or $\Gamma \longrightarrow \Delta, (\varphi \vee \psi)$. But the reversibility of rules and the specific form of the disjunction rules show that S is the union of the set of sequents $(\varphi \vee \psi), \Gamma \longrightarrow \Delta$ such that $\varphi, \Gamma \longrightarrow \Delta$

22 [13] introduces a different approach to proving completeness of sequent systems directly, based on the formalisation of the semantic theory, which could also be used for the present system.

23 Quantified formulas have an infinite number of subformulas, so the argument does not extend to predicate logic.

and $\psi, \Gamma \longrightarrow \Delta$ are valid and set of sequents $\Gamma \longrightarrow \Delta, (\varphi \vee \psi)$ such that $\Gamma \longrightarrow \Delta, \varphi, \psi$ is valid. The inferential meaning of $(\varphi \vee \psi)$ is therefore entirely determined by the inferential meaning of φ and ψ in the manner shown by the rules of G, as promised earlier, thus justifying the claim that the sequent calculus G is an analytic calculus for predicate logic.[24]

4 A Sequent Calculus for Aggregation

Our formulation of a sequent calculus for the logic of aggregation uses labelled formulas. These are expressions of the form $k{:}\,\varphi$, where φ is a formula in our language $\mathcal{L}_{\mathcal{AG}}$ and k is a nominal. A sequent calculus for these expressions is given in Figure 5. The structural rules and the rules for the operators of propositional logic track those of G, with the labels playing no role at all. The rules for modal and hybrid operators use the labels in a way that mirrors the truth conditions for these operators in their semantic theory.[25]

Additional rules for the modal operators respect their interpretation as preference operators, ensuring that the accessibility relation of $\langle t\rangle$ is a preorder and that of $\langle t\rangle^{<}$ is the corresponding strict preorder. Finally, the rules for aggregation again track the semantic conditions for the two aggregation operators.

To prove Cut-elimination, it is sufficient to show both Cut reduction and the permutation of all rules with Cut. For the latter, we note that the only non-context-independent rules are $\langle t\rangle$L and EL, which are context-sensitive, and :L, which is context-altering. But these are sufficiently similar to the rules $\exists$L and $=$L from predicate logic to be treated in the same way. Cut reduction must be considered for each pair of rules with matching left and right principal formulas. Here we give one examples of the required transformations.

$$\dfrac{\dfrac{\overset{\pi_1}{\Gamma \longrightarrow \Delta, k{:}\,\langle t_1\rangle i} \quad \overset{\pi_2}{\Gamma \longrightarrow \Delta, k{:}\,\langle t_2\rangle i}}{\Gamma \longrightarrow \Delta, k{:}\,\langle t_1 \parallel t_2\rangle i}\,\|\text{R} \qquad \dfrac{\overset{\pi_3}{k{:}\,\langle t_1\rangle i, k{:}\,\langle t_2\rangle i, \Gamma' \longrightarrow \Delta'}}{k{:}\,\langle t_1 \parallel t_2\rangle i, \Gamma' \longrightarrow \Delta'}\,\|\text{L}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\text{C}$$

$$\rightsquigarrow \quad \dfrac{\overset{\pi_2}{\Gamma \longrightarrow \Delta, k{:}\,\langle t_2\rangle i} \qquad \dfrac{\overset{\pi_1}{\Gamma \longrightarrow k{:}\,\langle t_2\rangle i} \quad \overset{\pi_3}{k{:}\,\langle t_1\rangle i, k{:}\,\langle t_2\rangle i, \Gamma' \longrightarrow \Delta'}}{k{:}\,\langle t_2\rangle i, \Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\text{C}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\text{C}$$

This is sufficient to establish Cut-elimination, using the argument described in Section 3.

[24] The term ‘analytic proof theory’ was coined by Smullyan in [14].

[25] Up to this point, the calculus is identical to a similar labelled calculus in [13]. The labels can be removed by adding the satisfaction operator @ to the language and performing additional steps of internalization.

Structural Rules	
I	$\Rightarrow \quad k{:}\,\varphi, \Gamma \longrightarrow \Delta, k{:}\,\varphi.$
S	$\Gamma \longrightarrow \Delta \quad \Rightarrow \quad \Gamma' \longrightarrow \Delta'$ if $\Gamma \approx \Gamma'$ and $\Delta \approx \Delta'$.

Propositional Rules	
¬L	$\Gamma \longrightarrow \Delta, k{:}\,\varphi \quad \Rightarrow \quad k{:}\,\neg\varphi, \Gamma \longrightarrow \Delta.$
¬R	$k{:}\,\varphi, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, \neg k{:}\,\varphi.$
∨L	$k{:}\,\varphi, \Gamma \longrightarrow \Delta;\ k{:}\,\psi, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad k{:}\,(\varphi \vee \psi), \Gamma \longrightarrow \Delta.$
∨R	$\Gamma \longrightarrow \Delta, k{:}\,\varphi, k{:}\,\psi \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,(\varphi \vee \psi).$

Modal Rules	
$\langle t\rangle$L	$i{:}\,\varphi, k{:}\,Mi, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad k{:}\,M\varphi, \Gamma \longrightarrow \Delta$ if i does not occur in $k, \varphi, \Gamma, \Delta$.
$\langle t\rangle$R	$\Gamma \longrightarrow \Delta, k{:}\,M\varphi, k{:}\,Mi;\ \Gamma \longrightarrow \Delta, k{:}\,M\varphi, i{:}\,\varphi \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,M\varphi.$
EL	$i{:}\,\varphi, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad k{:}\,E\varphi, \Gamma \longrightarrow \Delta$ if i does not occur in $k, \varphi, \Gamma, \Delta$.
ER	$\Gamma \longrightarrow \Delta, k{:}\,E\varphi, i{:}\,\varphi \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,E\varphi.$

Hybrid Rules	
$:\mathrm{L}_1$	$k{:}\,i, \Gamma[^{j}_{k}] \longrightarrow \Delta[^{j}_{k}] \quad \Rightarrow \quad k{:}\,i, \Gamma[^{j}_{i}] \longrightarrow \Delta[^{j}_{i}].$
$:\mathrm{L}_2$	$k{:}\,i, \Gamma[^{j}_{i}] \longrightarrow \Delta[^{j}_{i}] \quad \Rightarrow \quad k{:}\,i, \Gamma[^{j}_{k}] \longrightarrow \Delta[^{j}_{k}].$
$:\mathrm{R}$	$\Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,k\ .$

Preference Rules	
Re	$\Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,\langle t\rangle k\ .$
Tr	$\Gamma \longrightarrow \Delta, k{:}\,\langle t\rangle i;\ \Gamma \longrightarrow \Delta, i{:}\,\langle t\rangle j \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,\langle t\rangle j.$
<L	$k{:}\,\langle t\rangle i, \Gamma \longrightarrow \Delta, i{:}\,\langle t\rangle k \quad \Rightarrow \quad k{:}\,\langle t\rangle^{<} i, \Gamma \longrightarrow \Delta.$
<R	$\Gamma \longrightarrow \Delta, k{:}\,\langle t\rangle i;\ i{:}\,\langle t\rangle k, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,\langle t\rangle^{<} i.$

Aggregation Rules	
‖L	$k{:}\,\langle t_1\rangle i, k{:}\,\langle t_2\rangle i, \Gamma \longrightarrow \Delta \quad \Rightarrow \quad k{:}\,\langle t_1 \parallel t_2\rangle i, \Gamma \longrightarrow \Delta.$
‖R	$\Gamma \longrightarrow \Delta, k{:}\,\langle t_1\rangle i;\ \Gamma \longrightarrow \Delta, k{:}\,\langle t_2\rangle i \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,\langle t_1 \parallel t_3\rangle i.$
/L	$k{:}\,\langle t_1\rangle i, k{:}\,\langle t_2\rangle i, \Gamma \longrightarrow \Delta;\ k{:}\,\langle t_1\rangle i, \Gamma \longrightarrow \Delta, i{:}\,\langle t_1\rangle k$ $\Rightarrow \quad k{:}\,\langle t_1/t_2\rangle i, \Gamma \longrightarrow \Delta.$
/R	$\Gamma \longrightarrow \Delta, k{:}\,\langle t_1\rangle i;\ i{:}\,\langle t_1\rangle k, \Gamma \longrightarrow \Delta, k{:}\,\langle t_2\rangle i \quad \Rightarrow \quad \Gamma \longrightarrow \Delta, k{:}\,\langle t_1/t_2\rangle i.$

Fig. 5. A labelled sequent calculus AG for the logic of aggregation, with M ranging over $\langle t\rangle$ and $\langle t\rangle^{<}$ for each term t

Theorem 3. *Cut and Weakening are admissible in AG.*

We note that AG has the Subformula Property, given a suitable definition of 'subformula' according to which the subformulas of $\langle t\rangle\varphi$ are φ and $\langle t\rangle u$ for each nominal u; those of $\langle t\rangle^{<}\varphi$ are the same, with the addition of $\langle t\rangle^{<} v$ for each v; those of $\langle t_1 \parallel t_2\rangle\varphi$ and $\langle t_1/t_2\rangle\varphi$ are again the same as for all formulas formed using a modal operator, with in addition $\langle t_1\rangle\varphi$ and $\langle t_2\rangle\varphi$. $E\varphi$ has subformulas

φ and each nominal. The rules are all reversible and so the calculus is fully analytic, in the sense explained in Section 3.

By saying that AG is sound, we mean that if k is a nominal that does not occur in the $\mathcal{L}_{\mathcal{AG}}$ sequent $\Gamma \longrightarrow \Delta$ and $k\colon \Gamma \longrightarrow k\colon \Delta$ is provable in AG then $\Gamma \longrightarrow \Delta$ is valid. This can be established by inspection of the rules, in the usual way. To show the converse, completeness, by the Axioms to Rules Method, it is enough to show that the axioms and rules of the system in Figure 3 are derivable with the help of Weakening and Cut, which is a routine exercise. As a quick example, the following shows the derivability of rule 7, $\vdash i \rightarrow \varphi \quad \Rightarrow \quad \vdash \varphi$ (assuming i not in φ):

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\longrightarrow k\colon k}\,{:}\mathrm{R}}{\longrightarrow k\colon Ek}\,E\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{}{\longrightarrow k\colon k}\,{:}\mathrm{R} \quad \dfrac{\dfrac{\dfrac{\longrightarrow k\colon(\neg i \vee \varphi) \qquad \dfrac{\dfrac{\dfrac{}{k\colon i \longrightarrow k\colon i, k\colon \varphi}\,\mathrm{I}}{k\colon \neg i, k\colon i \longrightarrow k\colon \varphi}\,\neg\mathrm{L} \quad \dfrac{}{k\colon \varphi, k\colon i \longrightarrow k\colon \varphi}\,\mathrm{I}}{k\colon(\neg i \vee \varphi), k\colon i \longrightarrow k\colon \varphi}\,\vee\mathrm{L}}{k\colon i \longrightarrow k\colon \varphi}\,\mathrm{C}}{i\colon k, k\colon i \longrightarrow k\colon \varphi}\,\mathrm{W}}{i\colon k, k\colon k \longrightarrow k\colon \varphi}\,{:}\mathrm{L}_1}{i\colon k \longrightarrow k\colon \varphi}\,\mathrm{C}}{k\colon Ek \longrightarrow k\colon \varphi}\,E\mathrm{L}}}{\longrightarrow k\colon \varphi}\,\mathrm{C}$$

A couple of features of derivations like this one are worthy of comment. First, note that we re-write $i \rightarrow \varphi$ as $(\neg i \vee \varphi)$. This is necessary only because we limited the Boolean part of our sequent calculus to rules for negation and disjunction. With the standard definitions of the other connectives, their standard sequent calculus rules for them are all derivable, and so could be used here. This is a trivial point. Second, and more significantly, the derivation of rule 7 uses the existential modality, E, so providing a further insight into why E is needed in the axiomatic system. Without E, or something similar, hybrid logic is hard to axiomatize.[26] But when translated into the sequent calculus AG, we see how the occurrences of E disappear with Cut-elimination: by the subformula property of AG, the proof of any sequent not containing E will also not contain E.

Theorem 4. *If k is a nominal that does not occur in the $\mathcal{L}_{\mathcal{AG}}$ sequent $\Gamma \longrightarrow \Delta$ then $k\colon \Gamma \longrightarrow k\colon \Delta$ is provable in AG iff $\Gamma \longrightarrow \Delta$ is valid.*

The structure of AG allows us to generalise Theorem 4 considerably. By the subformula property of AG, any sequent in a fragment of $\mathcal{L}_{\mathcal{AG}}$ restricted to a subset of the logical operators will have a proof in the sequent calculus obtained by restricting AG to the rules that mention those operators, which will therefore be complete for the fragment. For example, if we simply drop the rules for E,

[26] An approach, using infinite (non-schematic) sets of axioms was given in [7].

we obtain a calculus for the E-free fragment, something that is very difficult to axiomatize, as noted above.

In conclusion, we have shown that a simple extension of $\mathcal{L}_{\mathcal{P}}$, the basic logic of preference for individuals, with nominals gives a complete logic for the lexicographic re-ordering of preferences given in [1]. We have shown two approaches to the proof theory of aggregation logic, one using a Hilbert-style axiomatization and the other using a Gentzen-style sequent calculus. For the former, our completeness result relies on the presence of the existential modality. For the latter, our rules mirror the inductive definition of terms found inside the modal operators and so there is no need for the existential modality. The Gentzen-style calculus yields an explicit analysis of group modalities into individual modalities.

Our logic is well suited for lexicographic re-ordering of preference relations, but it is quite clear that a similar approach could be taken with other aggregation operators. For instance, one might define an aggregation operator which follows the choices of subordinate agents when their boss abstains from voting. Lexicographic re-ordering fails to yields a result in such cases. Several questions arise: which properties, analogical to $IBUT$, does this operator satisfy? And what class of aggregation operators can be modeled using our approach?

References

1. Andréka, H., Ryan, M., Schobbens, P.-Y.: Operators and laws for combining preference relations. Journal of logic and computation 12(1), 13–53 (2002)
2. Areces, C., ten Cate, B.: Hybrid logic. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic. Elsevier, Amsterdam (2007)
3. Arrow, K.: Social choice and individual value. Wiley, Chichester (1951)
4. Boutilier, C.: Toward a logic for qualitative decision theory. In: Doyle, J., Sandewall, E., Torasso, P. (eds.) Principles of Knowledge Representation and Reasoning, pp. 75–86 (1994)
5. Brandom, R.B.: Making it explicit. Harvard university press (1994)
6. Dummett, M.: The logical basis of metaphysics. Harvard university press (1991)
7. Gargov, G., Passy, S., Tinchev, T.: Modal environment for boolean speculations (preliminary report). In: Skordev, D. (ed.) Mathematical Logic and its Applications. Proceedings of the Summer School and Conference dedicated to the 80th Anniversary of Kurt Gödel, pp. 253–263. Plenum Press, Druzhba (1987)
8. Gentzen, G.: Untersuchungen über das logische schließen. Mathematische Zeitschrift (39), 405–431 (1934)
9. Girard, P.: Modal logic for lexicographic preference aggregation (to appear, 2008)
10. Lang, J., van der Torre, L., Weydert, E.: Hidden uncertainty in the logical representation of desires. In: Proceedings of Eighteenth International Joint Conference on Artificial Intelligence, IJCAI 2003 (2003)
11. Restall, G.: An introduction to substructural logics. Routledge (2000)
12. Segerberg, K.: The basic dynamic doxastic logic of AGM. In: Williams, M.-A., Rott, H. (eds.) Frontiers in belief revision, pp. 57–84. Kluwer, Dordrecht (2001)
13. Seligman, J.: Internalization: The case of hybrid logics. Journal of logic and computation 11(5), 671–689 (2001)
14. Smullyan, R.M.: First-order logic. Dover (1995)

15. ten Cate, B.: Model Theory for Extended Modal Languages. PhD thesis, University of Amsterdam, ILLC Dissertation Series DS-2005-01 (2005)
16. Troelstra, A.S., Schwichtenberg, H.: Basic proof theory, 2nd edn. Cambridge tracts in theoretical computer science, vol. 43. Cambridge University Press, Cambridge (2000)
17. van Benthem, J., Girard, P., Roy, O.: Everything else being equal: a modal logic for *ceteris paribus* preferences. Journal of philosophical logic (August 2008)
18. van Benthem, J., van Otterloo, S., Roy, O.: Preference logic, conditionals, and solution concepts in games. In: Lagerlund, H., Lindström, S., Sliwinski, R. (eds.) Modality Matters: Twenty-Five Essays in Honour of Krister Segerberg. Uppsala Philosophical Studies, Uppsala (2006)

Instantial Relevance in Polyadic Inductive Logic

Jürgen Landes⋆, Jeff Paris, and Alena Vencovská⋆⋆

School of Mathematics
University of Manchester
Manchester M13 9PL, UK
juergen_landes@yahoo.de
{jeff.paris,alena.vencovska}@manchester.ac.uk

Abstract. We show that under the assumptions of Spectrum Exchangeability and Language Invariance the so called *Only Rule*, a principle of instantial relevance previously know for unary (i.e. classical) Carnapian Inductive Logic, also holds in Polyadic Inductive Logic.

Keywords: Instantial Relevance, Spectrum Exchangeability, Inductive Logic, Probability Logic, Uncertain Reasoning.

1 Introduction

In this paper we are interested in the following question: Given that we make a rational or logical assignment of probabilities to sentences according to the dictates of Inductive Logic under what conditions on sentences θ and ϕ *must* the probability of θ be at least that of ϕ?

In order to formalize this problem more precisely we first need to introduce some notation and notions from Inductive Logic. Let L be a first order language containing finitely many relation symbols $P_1, P_2, \ldots, P_q$ with arities $r_1, r_2, \ldots, r_q$ respectively, countably many constants $a_1, a_2, a_3, \ldots$ (the implicit intention being that these exhaust the universe) and no function symbols nor equality. Let SL, $QFSL$, respectively, denote the sentences and quantifier free sentences of L. Throughout we shall use $b_1, b_2, \ldots$ and $b'_1, b'_2, \ldots$ to denote distinct constants a_i from L.

A function $w : SL \to [0,1]$ is a *probability function* on L if it satisfies that for all $\theta, \phi, \exists x\, \psi(x) \in SL$:

(P1) If $\vDash \theta$ then $w(\theta) = 1$.

(P2) If $\vDash \neg(\theta \wedge \phi)$ then $w(\theta \vee \phi) = w(\theta) + w(\phi)$.

(P3) $w(\exists x\, \psi(x)) = \lim_{n\to\infty} w(\bigvee_{i=1}^{n} \psi(a_i))$.

From this (P1-3) all the expected properties (see for example [15, page 10]) of 'probability' follow, for example

⋆ Supported by a MATHLOGAPS Research Studentship - MEST-CT-2004-504029.

⋆⋆ Supported by a UK Engineering and Physical Sciences Research Council (EPSRC) Research Assistantship.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 162–169, 2009.

(P4) If $\models \neg\theta$ then $w(\theta) = 0$,

(P5) If $\phi \models \theta$ then $w(\phi) \leq w(\theta)$.

(P5) would appear to give a partial answer to the question we asked at the start, indeed it is easy to show that the converse holds too, namely, $w(\theta) \geq w(\phi)$ for all probability functions on L if and only if $\phi \models \theta$.

This however is far from the end of the matter. For in Inductive Logic we are not interested in *all* probability functions on L but only those probability functions which are 'rational' or 'logical', which in this context is identified with satisfying certain principles which we intuitively feel a rational agent should abide by when assigning probabilities, see for example [1], [2], [3], [8]. There are now a number of such principles (in addition to the above see for example [6], [15], [16]) but as far as this paper is concerned we shall be interested in just two of them, Spectrum Exchangeability, Sx, and Language Invariance, LI. In order to explain these we need to develop a little more notation.

By a theorem of Gaifman (see [4]) any probability function defined on $QFSL$ (i.e. satisfying (P1) and (P2) for $\theta, \phi \in QFSL$) extends uniquely to a probability function on L. Hence we can limit our considerations to probability functions defined just on $QFSL$. By the Disjunctive Normal Form Theorem it then follows that w is determined simply by its values on the *state descriptions*, that is sentences of the form $\Theta(b_1, b_2, \ldots, b_m)$ where

$$\Theta(b_1, b_2, \ldots, b_m) \quad = \quad \bigwedge_{s=1}^{q} \quad \bigwedge_{i_1, i_2, \ldots, i_{r_s} \in \{1, \ldots, m\}} \pm P_s(b_{i_1}, b_{i_2}, \ldots, b_{i_{r_s}}) \tag{1}$$

and $\pm P$ stands for one of P or $\neg P$ in each case.

Given a state description $\Theta(b_1, \ldots, b_m)$, we define an equivalence $\sim_\Theta$ on $\{1, 2, \ldots, m\}$ as follows:[1]

$i \sim_\Theta j$ if whenever $\Theta'(b_1, b_2, \ldots, b_m)$ is obtained from $\Theta(b_1, b_2, \ldots, b_m)$ by replacing some of the occurrences of b_i by b_j and/or some of the occurrences b_j by b_i then $\Theta'(b_1, b_2, \ldots, b_m)$ is consistent with $\Theta(b_1, b_2, \ldots, b_m)$. Putting it another way $i \sim_\Theta j$ if

$$\Theta(b_1, \ldots, b_m) \wedge b_i = b_j$$

is consistent with the axioms of equality (for the language L with $=$ added).

Clearly $\sim_\Theta$ is an equivalence relation. Let the sizes of the equivalence classes of $\sim_\Theta$ be $n_1, n_2, \ldots, n_r$ in decreasing order of size. We define the *Spectrum* $\mathcal{S}(\Theta)$ of this state description $\theta(b_1, \ldots, b_m)$ to be the vector $\langle n_1, n_2, \ldots, n_r \rangle$.

We are now ready to state:

The Spectrum Exchangeability Principle (Sx)

If $\Theta(b_1, b_2, \ldots, b_m), \Phi(b'_1, b'_2, \ldots, b'_m)$ *are state descriptions and* $\mathcal{S}(\Theta) = \mathcal{S}(\Phi)$ *then*

$$w(\Theta(b_1, b_2, \ldots, b_m)) = w(\Phi(b'_1, b'_2, \ldots, b'_m)).$$

[1] This definition depends on the order $b_1, \ldots, b_m$ we give to the constants. However this dependence will become irrelevant when we come to defining Spectrum Exchangeability, Sx.

This means that w of a state description $\Theta(b_1, b_2, \ldots, b_m)$ depends only on the spectrum $\mathcal{S}(\Theta)$, not on any other specific property of Θ nor on the particular constants $b_1, b_2, \ldots, b_m$. In short, one justification for this principle as a reflection of rationality is that knowing nothing about the individual relations the only reason we might have for giving different probabilities to the state descriptions $\Theta(b_1, b_2, \ldots, b_m)$, $\Phi(b'_1, b'_2, \ldots, b'_m)$ is because they have different spectra. [The original, and in fact different, justification for Sx is given in [14].]

As a principle it generalizes a number of other 'rationality principles' for example Constant Exchangeability, Ex, and Atom Exchangeability (a principle in the original Carnapian, or unary, Inductive Logic, see [15] for example). It has a number of nice consequences, for short surveys see [9], [10], [14].

A second principle that we might feel it is natural to impose is that adding a further finitely many relations to the language L to give L^+ should not have any effect as far as probabilities of sentences in SL are concerned. After all there is no reason why L should from the start contain all the relations there could ever be. Given the aforementioned desirability of Sx this leads to the requirement that w satisfy:

Language Invariance, LI, (with Sx)
The probability function w on L satisfies LI if there is a family of probability functions $w_{L'}$ on languages L' (as above) satisfying Sx such that $w = w_L$ and whenever L'' extends L' then $w_{L''}$ agrees with $w_{L'}$ on SL'.

We are now almost ready to state the main result of this paper, an answer to our initial problem in the case w satisfies[2] Sx + LI and θ, ϕ are *state descriptions of the same length*, i.e. for the same number of constants. First though we need to define an ordering on spectra.

Given spectra $\boldsymbol{n} = \langle n_1, n_2, \ldots, n_r\rangle$ and $\boldsymbol{m} = \langle m_1, m_2, \ldots, m_t\rangle$ with $\sum_{i=1}^r n_i = \sum_{i=1}^t m_i$ we define

$$\boldsymbol{m} \preceq \boldsymbol{n} \iff \sum_{i\leq j} m_i \leq \sum_{i \leq j} n_i \quad \text{for all } j = 1, 2, \ldots, \max\{r, t\}$$

where we take $n_i = 0$ for $r < i \leq \max\{r,t\}$ and $m_i = 0$ for $t < i \leq \max\{r,t\}$.

So for example $\langle 5,3,2,1\rangle \preceq \langle 5,5,1\rangle$ since $5 \leq 5,\ 5+3 \leq 5+5,\ 5+3+2 \leq 5+5+1,\ 5+3+2+1 \leq 5+5+1+0$.

Theorem 1. *Let $\Theta(b_1, \ldots, b_m)$, $\Phi(b'_1, b'_2, \ldots, b'_m)$ be state descriptions (for L). Then $w(\Phi) \leq w(\Theta)$ for all probability functions w on L satisfying Sx + LI if and only if $\mathcal{S}(\Phi) \preceq \mathcal{S}(\Theta)$.*

Proof. We first need to define a particular family of probability functions satisfying Sx + LI which are in some sense the building blocks for all probability functions satisfying these principles.

[2] Of course the Sx is really redundant here because it is included in our definition of LI.

Let

$$\mathbb{B} = \left\{ \langle x_0, x_1, x_2, \ldots \rangle \,|\, x_1 \geq x_2 \geq \ldots \geq 0,\, x_0 \geq 0,\, \sum_{i=0}^{\infty} x_i = 1 \right\}$$

and endow $\mathbb{B}$ with the standard weak product topology inherited from $[0,1]^{\infty}$. Let

$$\overline{p} = \langle p_0, p_1, p_2, \ldots \rangle \in \mathbb{B}.$$

We shall be thinking of p_i as the probability of picking 'colour' i (from some urn, with replacement).

For a state description $\Theta(b_1, b_2, \ldots, b_m)$ and a sequence of colours

$$\boldsymbol{c} = \langle c_1, c_2, \ldots, c_m \rangle \in \{0, 1, 2, \ldots\}^m$$

(where 0 stands for the special colour black) we define probabilities

$$j^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_m), \boldsymbol{c})$$

inductively as follows:

Set $j^{\overline{p}}(\top, \emptyset) = 1$. Suppose that at stage m we have defined the probability $j^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_m), \boldsymbol{c})$. Pick colour c_{m+1} from $0, 1, 2, \ldots$ according to the probabilities $p_0, p_1, p_2, \ldots$ and let

$$\boldsymbol{c}^+ = \langle c_1, \ldots, , c_m, c_{m+1} \rangle.$$

If c_{m+1} is the same as an earlier colour, c_j say, with $c_j \neq 0$ extend $\Theta(b_1, b_2, \ldots, b_m)$ to the unique state description $\Theta^+(b_1, b_2, \ldots, b_m, b_{m+1})$ for which $b_j \sim_{\Theta^+} b_{m+1}$. (Notice this means that the equivalence classes mod Θ^+ are the same as those mod Θ except that $m+1$ is added to the class containing j.) On the other hand if c_{m+1} is 0 or a previously unchosen colour then randomly choose $\Theta^+(b_1, b_2, \ldots, b_m, b_{m+1})$ extending $\Theta(b_1, b_2, \ldots, b_m)$ so that when $i, j \leq q$ are such that $c_i = c_j \neq 0$ then $b_i \sim_{\Theta^+} b_j$ (where 'randomly' means that we take all possibilities with equal probability). Finally let $j^{\overline{p}}(\Theta^+, \boldsymbol{c}^+)$ be $j^{\overline{p}}(\Theta, \boldsymbol{c})$ times the probability as described of going from $\Theta, \boldsymbol{c}$ to $\Theta^+, \boldsymbol{c}^+$.

Having defined these $j^{\overline{p}}(\Theta, \boldsymbol{c})$ now set

$$u^{\overline{p}}(\Theta) = \sum_{\boldsymbol{c}} j^{\overline{p}}(\Theta, \boldsymbol{c}).$$

By a straightforward generalization of the result in [14] (where just two colours were considered) $u^{\overline{p}}$ satisfies Sx (and hence also Ex) and by results in [12] $u^{\overline{p}}$ satisfies LI. Notice that an important consequence of $u^{\overline{p}}$ satisfying Sx is that when considering $u^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_m))$ for $\Theta(b_1, b_2, \ldots, b_m)$ a state description we can assume that the equivalence classes with respect to $\sim_\Theta$ are consecutive blocks $1, \ldots, n_1$, $n_1 + 1, \ldots, n_1 + n_2$, $\ldots$ from $1, 2, \ldots, m$, which will be a help in visualizing the forthcoming expansion of $u^{\overline{p}}(\Theta)$ as a sum of products.

It now follows that when μ is a probability measure on $\mathbb{B}$ and we define

$$w(\Theta) = \int_{\mathbb{B}} u^{\overline{p}}(\Theta) d\mu(\overline{p}) \tag{2}$$

then w is a probability function on SL satisfying Sx + LI. Somewhat less obviously the converse is also true, namely *any* probability function w on L satisfying Sx + LI is of this form for some measure μ on $\mathbb{B}$.

This result is the main Theorem 8 of [11]. Its importance to us here is that it is enough to show Theorem 1 in the case where $w = u^{\overline{p}}$, the general (right to left) implication following by then taking an integral over $\mathbb{B}$ with respect to some measure μ. In fact we will first show that it is even sufficient to show it in the case where $p_0 = 0$ in $\overline{p}$, in other words when there is zero probability of picking black.

To see this consider a sequence of colours $\boldsymbol{c} = \langle c_1, c_2, \ldots, c_m \rangle$ leading to a contribution $j^{\overline{p}}(\Theta, \boldsymbol{c})$ to $u^{\overline{p}}(\Theta)$. Let N be large and let $\overline{q}$ be the vector resulting from $\overline{p}$ by replacing the colour black by N shades of grey, each assigned probability p_0/N. In other words $q_0 = 0$ and the q_i for $i > 0$ are just the same probabilities $p_1, p_2, p_3, \ldots$ of the old non-black colours together with N probabilities $p_0/N, p_0/N, \ldots, p_0/N$ for these new greys. Then

$$j^{\overline{p}}(\Theta, \boldsymbol{c}) = \sum_{\boldsymbol{k}} \underline{j}^{\overline{q}}(\Theta, \boldsymbol{k}) \tag{3}$$

where the $\boldsymbol{k}$ are all choices of colours which agree with $\boldsymbol{c}$'s colours (and probabilities) when not black and allow any choice of a grey (each with probability p_0/N) when $\boldsymbol{c}$ picks black, *and* $\underline{j}$ is defined like j except that when the colour is a gray we just pick the extending state description at random irrespective of whether that grey had actually already been chosen. In other words $\underline{j}$ treats these greys just the same way j treated the black.

Now consider

$$\left| \sum_{\boldsymbol{k}} \underline{j}^{\overline{q}}(\Theta, \boldsymbol{k}) - \sum_{\boldsymbol{k}} j^{\overline{q}}(\Theta, \boldsymbol{k}) \right|.$$

The summands which appear in one of these but not the other are those in the first sum for which the same grey colour was chosen (at least) twice, say as k_i, k_j but $i \not\sim_{\Theta} j$. However, even if we ignore that second condition the probability as we pick $k_1, k_2, \ldots, k_m$ of ever picking any one of the N greys at least twice is at most $N \times (1/N)^2 \times {}^m C_2 \leq m^2/2N$. It follows then that

$$\lim_{N \to \infty} u^{\overline{q}}(\Theta) = u^{\overline{p}}(\Theta).$$

From this it follows that in order to prove our theorem it is enough to prove it for a $u^{\overline{p}}$ with $p_0 = 0$.

So assume that $p_0 = 0$ and for the right to left direction of the theorem let

$$\mathcal{S}(\Theta(b_1, b_2, \ldots, b_m)) = \langle n_1, n_2, \ldots, n_r \rangle = \boldsymbol{n}$$

and

$$\mathcal{S}(\Phi(b'_1, b'_2, \ldots, b'_m)) = \langle m_1, m_2, \ldots, m_r \rangle = \boldsymbol{m},$$

where if necessary we have appended zero's to make both vectors the same length, and suppose that $\boldsymbol{m} \preceq \boldsymbol{n}$. In this case, for this special $\overline{p}$, $u^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_m)$ equals

$$\sum_{k>0} \sum_{\substack{X \subset \mathbb{N}^+ \\ |X|=k}} \sum_{\substack{S_1 \cup \ldots \cup S_r = X \\ S_i \cap S_j = \emptyset,\ 1 \le i < j \le r}} G_{r,m} \prod_{j=1}^{r} \left(\sum_{i \in S_j} p_i \right)^{\Box n_j}, \tag{4}$$

where the $\Box$ in $(\sum_{i \in S_j} p_i)^{\Box n_j}$ etc. indicates that in the expansion of this power we only count those terms which have a non-zero power of p_i for each $i \in S_j$, etc., and $G_{r,m}$ is a probability factor corresponding to the random choices of sub-state descriptions with each new colour seen and depends only on k and r (and L).

Hence to show that $u^{\overline{p}}(\Phi) \le u^{\overline{p}}(\Theta)$ it is enough, by choosing a particular k and X and employing (4) for both Φ and Θ to show that

$$\sum_{\substack{S_1 \cup \ldots \cup S_r = X \\ S_i \cap S_j = \emptyset,\, 1 \le i < j \le r}} \prod_{j=1}^{r} \left(\sum_{i \in S_j} p_i \right)^{\Box m_j} \le \sum_{\substack{S_1 \cup \ldots \cup S_r = X \\ S_i \cap S_j = \emptyset,\, 1 \le i < j \le r}} \prod_{j=1}^{r} \left(\sum_{i \in S_j} p_i \right)^{\Box n_j}. \tag{5}$$

This half of the proof can now be completed by noting that a proof of (5) is given in [17].

To show the other direction suppose that $\Theta(b_1, b_2, \ldots, b_m), \Phi(b'_1, b'_2, \ldots, b'_m)$ are such that

$$\langle m_1, m_2, \ldots, m_t \rangle = \mathcal{S}(\Phi) \not\preceq \mathcal{S}(\Theta) = \langle n_1, n_2, \ldots, n_r \rangle,$$

with $m_t > 0$, and let $j \le t$ be such that

$$M = \sum_{i \le j} m_i > \sum_{i \le j} n_i = N.$$

Let $\overline{p}$ be such that $p_0 = 0 = p_i$ for $i > t$, $p_i = (1-\epsilon)/j$ for $i \le j$, and $p_i = \epsilon/(s-j)$ for $j < i \le s$, where $\epsilon > 0$ is small and $s = \max\{t, r\}$. Then it is straightforward to see that

$$u^{\overline{p}}(\Phi(b'_1, b'_2, \ldots, b'_m)) \ge d\epsilon^{m-M}$$

for some $d > 0$ and

$$u^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_m)) = O(\epsilon^{m-N}),$$

which, since $m - M < m - N$ through choosing ϵ sufficiently small, gives as required that

$$u^{\overline{p}}(\Phi(b'_1, b'_2, \ldots, b'_m)) \not\le u^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_m)).$$

Theorem 1 answers our question, *under what conditions on sentences θ and ϕ must the probability of θ be at least that of ϕ?* in the case when the probability

function is also required to satisfy Sx + LI and θ, ϕ are state descriptions of the same length. Whilst this might seem somewhat restrictive the result, and method of proof, actually allow some improvements, for example to deciding the situation between state descriptions $\Theta(b_1, b_2, \ldots, b_m)$ and $\Phi(b'_1, b'_2, \ldots, b'_k)$ when $m \neq k$ and in some cases for disjunctions of state descriptions. The completely general case (for w satisfying Sx + LI) however looks difficult and at the end of the day possibly not very illuminating.

2 Conclusion

Theorem 1 is an example of 'Instantial Relevance' in that it is one way of capturing the idea that *the more you have seen something in the past the more likely you should expect to see it in the future*. There are a number of other possible formulations here, in particular one proposed by Carnap for *unary* Inductive Logic is usually taken to be the bearer of this name *Principle of Instantial Relevance*, see [1, Chapter VI]. It was a pleasing discovery by Gaifman (see [4], or [7] for a subsequent somewhat simpler proof) that this principle for unary Inductive Logic followed from Constant Exchangeability, the mild assumption that w should be invariant under permutations of constants, itself an immediate consequence of Sx.

This original version of Instantial Relevance for unary languages does not seem to easily generalize to Polyadic Inductive Logic because it supposes that a state description involving an a_i fixes all there is to know about a_i, which is no longer true once we allow non-unary relations. The result for unary Inductive Logic which accurately corresponds to the formulation of Instantial Relevance considered in this paper was proved in [18], this version of Theorem 1 being there referred to as the 'Only Rule' because, as here, it was shown in that paper to be the only rule of that kind which holds, in that case for w satisfying Atom Exchangeability, the equivalent of Sx for unary languages.

As already implied however the above are but two ways of capturing the intuition that 'the more you have seen something in the past the more likely you should expect to see it in the future'. Generally however the situation, even for the purely unary case, remains puzzling. For example in [18] it is shown that certain seemingly innocuous variants of Carnap's Instantial Relevance do not hold in general, or, as shown in [13], may lead to somewhat unacceptable conclusions. It seems that we still have much to understand about 'relevance'.

References

1. Carnap, R.: A basic system of inductive logic, part 1. In: Carnap, R., Jeffrey, R.C. (eds.) Studies in Inductive Logic and Probability, vol. I, pp. 33–165. University of California Press, Berkeley (1971)
2. Carnap, R.: A basic system of inductive logic. In: Jeffrey, R.C. (ed.) Studies in Inductive Logic and Probability, vol. II, pp. 7–155. University of California Press (1980)

3. Fitelson, B.: Inductive Logic, http://fitelson.org/il.pdf
4. Gaifman, H.: Concerning measures on first order calculi. Israel Journal of Mathematics 2, 1–18 (1964)
5. Gaifman, H.: Applications of de Finetti's Theorem to inductive logic. In: Carnap, R., Jeffrey, R.C. (eds.) Studies in Inductive Logic and Probability, vol. I. University of California Press, Berkeley (1971)
6. Hill, M.J., Paris, J.B., Wilmers, G.M.: Some observations on induction in predicate probabilistic reasoning. Journal of Philosophical Logic 31(1), 43–75 (2002)
7. Humburg, J.: The principle of instantial relevance. In: Carnap, R., Jeffrey, R.C. (eds.) Studies in Inductive Logic and Probability, vol. I. University of California Press, Berkeley, Los Angeles (1971)
8. Kemeny, J.G.: Carnap's Theory of Probability and Induction. In: Schilpp, P.A. (ed.) The Philosophy of Rudolf Carnap, La Salle, Illinois, Open Court, pp. 711–738 (1963)
9. Landes, J., Paris, J.B., Vencovská, A.: The Principle of Conformity and Spectrum Exchangeability. In: Löwe, B., Romeijn, J.-W. (eds.) Foundations of Formal Sciences VI. Studies in Logic. College Publications (to appear, 2008)
10. Landes, J., Paris, J.B., Vencovská, A.: Some Aspects of Polyadic Inductive Logic. Studia Logica (submitted)
11. Landes, J., Paris, J.B., Vencovská, A.: Representation Theorems for probability functions satisfying Spectrum Exchangeability in Inductive Logic. Journal of Symbolic Logic (submitted)
12. Landes, J., Paris, J.B., Vencovská, A.: A characterization of the Language Invariant Families satisfying Spectrum Exchangeability in Polyadic Inductive Logic. In: Proceeding of the IPM Logic Conference, Tehran (to appear, 2007); in a special edition of the Annals of Pure and Applied Logic
13. Nix, C.J., Paris, J.B.: A Continuum of Inductive Methods arising from a Generalized Principle of Instantial Relevance. Journal of Philosophical Logic 35(1), 83–115 (2006)
14. Nix, C.J., Paris, J.B.: A note on binary inductive logic. Journal of Philosophical Logic 36(6), 735–771 (2007)
15. Paris, J.B.: The Uncertain Reasoner's Companion. Cambridge University Press, Cambridge (1994)
16. Paris, J.B.: Short course on Inductive Logic, JAIST 2007 (2007), http://www.maths.manchester.ac.uk/~jeff/
17. Paris, J.B., Vencovská, A.: A Generalization of Muirhead's Inequality. Journal of Mathematical Inequalities and Applications (submitted)
18. Paris, J.B., Waterhouse, P.: Atom exchangeability and instantial relevance. Journal of Philosophical Logic (submitted)

Algebraic Study of Lattice-Valued Logic and Lattice-Valued Modal Logic

Yoshihiro Maruyama

Faculty of Integrated Human Studies, Kyoto University, Japan
maruyama@i.h.kyoto-u.ac.jp

Abstract. In this paper, we study lattice-valued logic and lattice-valued modal logic from an algebraic viewpoint. First, we give an algebraic axiomatization of L-valued logic for a finite distributive lattice L. Then we define the notion of prime L-filters and prove an L-valued version of prime filter theorem for Boolean algebras, by which we show a Stone-style representation theorem for algebras of L-valued logic and the completeness with respect to L-valued semantics. By the representation theorem, we can show that a strong duality holds for algebras of L-valued logic and that the variety generated by L coincides with the quasi-variety generated by L. Second, we give an algebraic axiomatization of L-valued modal logic and establish the completeness with respect to L-valued Kripke semantics. Moreover, it is shown that L-valued modal logic enjoys finite model property and that L-valued intuitionistic logic is embedded into L-valued modal logic of **S4**-type via Gödel-style translation.

1 Introduction

In 1991, Fitting [10] introduced L-valued logic and L-valued modal logics for a finite distributive lattice L. In a series of papers ([10], [11], [12]), Fitting studied those logics from a proof-theoretic viewpoint and did not consider the algebraic aspects of them. We remark that all the elements of L are encoded as truth constants in the languages of the Fitting's logics. Based on Fitting's work, some authors revealed several model-theoretic properties of L-valued modal logics ([5],[13],[14],[15]). But none of them give algebraic axiomatizations of L-valued logic or L-valued modal logics. This paper is a first step in obtaining them.

In this paper, we study Fitting's L-valued logic and L-valued modal logic modified by removing fuzzy truth constants (other than $0, 1$) and adding new unary connectives T_a(-)'s for all $a \in L$. $\mathrm{T}_a(x)$ intuitively means that the truth value of x is exactly a.

Some of the motivations for the above modifications are as follows. The existence of a truth constant $a \in L$ with $a \neq 0, 1$ philosophically means that there is a propositon x such that the truth value of x is "always exactly" a, which contradicts our intuition, since the truth value of a fuzzy proposition may vary from one person to another, from one time to another or from one possible world to another. Hence we remove fuzzy truth constants (other than $0, 1$). Unlike fuzzy truth constants, T_a's do not have such ontological commitment and only refer

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 170–184, 2009.

to the truth values of propositions. It seems that there may be a expression having similar meaning to $\mathrm{T}_a(x)$ in our natural languages, though there may be no expression having similar meaning to fuzzy truth constants (other than $0, 1$) in them. Thus, T_a's seem to be natural connectives and therefore it should be significant to investigate the properties of T_a's from the viewpoint of mathematical logic. In fact, operations like T_a(-) are also considered in the context of Post algebras (for example, see [6]). Finally, it must be stressed that the above modifications have some technical advantages as is shown in this paper (for a duality result obtained from the modifications, see Remark 1).

In addition to several algebraic axiomatizations with completeness, we will show: (i) L-valued versions of prime filter theorem and Stone's representation theorem for Boolean algebras; (ii) the finite model property of L-valued modal logic and a Gödel-Tarski-McKinsey style theorem between L-valued modal logic of **S4**-type and L-valued intuitionistic logic. By the representation theorem and natural duality theory ([3]), we can obtain a strong duality for algebras of L-valued logic (for another duality, see Section 4).

Note that our definition of primeness (Definition 5) is different from the ordinary one (i.e., $x \vee y \in P$ implies $x \in P$ or $y \in P$), which does not work well for (i) above, but our definition actually does.

2 Lattice-Valued Logic *L*-VL

Throughout this paper L denotes a finite distributive lattice. Then, L is a finite Heyting algebra. For $a, b \in L$, $a \rightarrow_L b$ denotes the relative pseudo-complement of a with respect to b, where the subscript L is often dropped.

Definition 1. *We endow L with the unary operations T_a(-)'s for all $a \in L$, which are defined by, for $b \in L$,*

$$\mathrm{T}_a(b) = \begin{cases} 1 & (\textit{if } b = a) \\ 0 & (\textit{if } b \neq a) \end{cases}$$

We define L-valued logic L**-VL** from a semantical point of view. The logical connectives of L**-VL** are $\wedge$, $\vee$, $\rightarrow$, 0, 1 and T_a for each $a \in L$, where every T_a is a unary connective, 0 and 1 are nullary connectives, and the others are binary connectives. **PV** denotes the set of propositional variables. Then, the formulas of L**-VL** are recursively defined in a usual way. We denote by **Form** the set of formulas of L**-VL**.

Definition 2. *v is an L-valuation iff v is a function from* **Form** *to L and satisfies the following properties:*

$$\begin{aligned} v(\mathrm{T}_a(x)) &= \mathrm{T}_a(v(x)); \\ v(x \wedge y) &= \inf(v(x), v(y)); \\ v(x \vee y) &= \sup(v(x), v(y)); \\ v(x \rightarrow y) &= v(x) \rightarrow_L v(y); \\ v(a) &= a \quad \text{for } a = 0, 1. \end{aligned}$$

Then, x ($\in$ **Form**) is called a valid formula of L-**VL** iff $v(x) = 1$ for all L-valuations v. If L is the two-element Boolean algebra, the valid formulas of L-**VL** coincide with the ordinary tautologies of classical propositional logic ($\mathrm{T}_1(x)$ (resp. $\mathrm{T}_0(x)$) is equivalent to x (resp. $\neg x$)).

Given the definition of validity, we can see that L-**VL** is an algebraically determined logic. The naturally associated classes of algebras are the variety $\mathcal{V}(L)$ generated by L and the quasi-variety $\mathcal{Q}(L)$ generated by L, and from universal algebra ([2],[3],[4]) we have the following:

Proposition 1. *Let $x, y \in$ **Form**. The following are equivalent:*

(i) $x = y$ holds in every algebra in $\mathcal{V}(L)$;
(ii) $x = y$ holds in every algebra in $\mathcal{Q}(L)$;
(iii) $v(x) = v(y)$ for every L-valuation v;
(iv) $x = y$ holds in L.

*In particular, $x \in$ **Form** is valid iff the equation $x = 1$ holds in L.*

Since L is finite, it is decidable whether or not a formula is valid in L-**VL**.

2.1 An Algebraic Axiomatization of L-VL

It is well known that every (quasi-)variety is axiomatizable by a collection of (quasi-)equations. The main result in this section is a finite such axiomatization of $\mathcal{V}(L)$, which actually coincides with $\mathcal{Q}(L)$ (see Theorem 3). In the process, we identify an internal description of homomorphisms into L as prime L-filters (see Definitions 5 and 6, and Proposition 5), thus generalizing the representation theory for Boolean algebras (see Theorem 2 and Remark 1). Note that $A \in \mathcal{Q}(L)$ iff the homomorphisms from A to L separate the points of A.

Now we give an algebraic axiomatization of L-**VL**. $x \leq y$ denotes $x \wedge y = x$. $x \leftrightarrow y$ is the abbreviation of $(x \to y) \wedge (y \to x)$.

Definition 3. $(A, \wedge, \vee, \to, \mathrm{T}_a(a \in L), 0, 1)$ *is an L-**VL**-algebra iff it satisfies the following axioms:*

(i) $(A, \wedge, \vee, \to, 0, 1)$ *forms a Heyting algebra;*
(ii) $\mathrm{T}_a(x) \wedge \mathrm{T}_b(y) \leq \mathrm{T}_{a\to b}(x \to y) \wedge \mathrm{T}_{a\wedge b}(x \wedge y) \wedge \mathrm{T}_{a\vee b}(x \vee y)$,
$\mathrm{T}_b(x) \leq \mathrm{T}_{\mathrm{T}_a(b)}(\mathrm{T}_a(x))$;
(iii) $\mathrm{T}_0(0) = 1$, $\mathrm{T}_a(0) = 0$ *(for $a \neq 0$)*, $\mathrm{T}_1(1) = 1$, $\mathrm{T}_a(1) = 0$ *(for $a \neq 1$)*;
(iv) $\bigvee\{\mathrm{T}_a(x)\,;\, a \in L\} = 1$, $\mathrm{T}_a(x) \vee (\mathrm{T}_a(x) \to 0) = 1$,
$\mathrm{T}_a(x) \wedge \mathrm{T}_b(x) = 0$ *(for $a \neq b$)*;
(v) $\mathrm{T}_1(\mathrm{T}_a(x)) = \mathrm{T}_a(x)$, $\mathrm{T}_0(\mathrm{T}_a(x)) = \mathrm{T}_a(x) \to 0$, $\mathrm{T}_b(\mathrm{T}_a(x)) = 0$ *(for $b \neq 0, 1$)*;
(vi) $\mathrm{T}_1(x) \leq x$, $\mathrm{T}_1(x \wedge y) = \mathrm{T}_1(x) \wedge \mathrm{T}_1(y)$;
(vii) $\bigwedge_{a\in L}(\mathrm{T}_a(x) \leftrightarrow \mathrm{T}_a(y)) \leq x \leftrightarrow y$.

The inequality $\mathrm{T}_a(x) \wedge \mathrm{T}_b(y) \leq \mathrm{T}_{a\to b}(x \to y)$ intuitively means that if the truth value of x is a and the truth value of y is b then the truth value of $x \to y$ is $a \to b$. The other inequalities following from (ii) can be explained similarly.

Note that T_1 is order-preserving by the axiom $\mathrm{T}_1(x \wedge y) = \mathrm{T}_1(x) \wedge \mathrm{T}_1(y)$.

By the axiom (i), we have: $x \leftrightarrow y = 1$ iff $x = y$.

We call the axiom $\bigvee\{\mathrm{T}_a(x)\,;\, a \in L\} = 1$ the L-valued excluded middle, since the two-valued excluded middle coincides with the ordinary excluded middle, which is shown in the proof of the following proposition.

Proposition 2. *Let L be the two-element Boolean algebra. Then, L-***VL***-algebras coincide with Boolean algebras.*

Proof. We claim that $\mathrm{T}_1(x) = x$ and $\mathrm{T}_0(x) = \mathrm{T}_1(x) \to 0$. Since $\mathrm{T}_1(x) \wedge \mathrm{T}_0(x) \leq 0$ by the axiom $\mathrm{T}_a(x) \wedge \mathrm{T}_b(x) = 0$, we have the inequality $\mathrm{T}_0(x) \leq \mathrm{T}_1(x) \to 0$. Assume that $\mathrm{T}_1(x) \wedge y \leq 0$. Then, by the axiom $\bigvee\{\mathrm{T}_a(x)\,;\, a \in L\} = 1$, we have $y \wedge (\mathrm{T}_1(x) \vee \mathrm{T}_0(x)) = y$. Thus, $y \wedge \mathrm{T}_0(x) = y$, i.e., $y \leq \mathrm{T}_0(x)$. Therefore, $\mathrm{T}_0(x) = \mathrm{T}_1(x) \to 0$. Thus we have $\mathrm{T}_0(\mathrm{T}_1(x)) = \mathrm{T}_1(\mathrm{T}_1(x)) \to 0 = \mathrm{T}_1(x) \to 0 = \mathrm{T}_0(x)$. Moreover, we have $\mathrm{T}_1(\mathrm{T}_1(x)) = \mathrm{T}_1(x)$. Therefore, by using the axiom (vii), we can conclude that $\mathrm{T}_1(x) = x$. Hence, the L-valued excluded middle coincide with the excluded middle. The remaining part of the proof is trivial.

We consider the various concepts of filters, which are essentially used when proving a representation theorem and completeness theorems.

Definition 4. *Let A be an L-***VL***-algebra and F a non-empty proper subset of A. Then, F is an L-filter (or L-valued filter) of A iff the following hold:*

(i) *if $x \in F$ and $x \leq y$ then $y \in F$;*
(ii) *if $x, y \in F$ then $x \wedge y \in F$;*
(iii) *if $x \in F$ then $\mathrm{T}_1(x) \in F$.*

Definition 5. *Let P be an L-filter of an L-***VL***-algebra A.*

(i) *P is a prime L-filter of A iff, for any $c \in L$, $\mathrm{T}_c(x \vee y) \in P$ implies that there exist $a, b \in L$ with $a \vee b = c$ such that $\mathrm{T}_a(x) \in P$ and $\mathrm{T}_b(y) \in P$.*
(ii) *P is an ultra L-filter of A iff $\forall x \in A\ \exists a \in L\ \mathrm{T}_a(x) \in P$.*
(iii) *P is a maximal L-filter iff P is maximal by inclusion.*

If L is the two-element Booelan algebra, then L-filters, prime L-filters, and ultra L-filters coincide with filters, prime filters, and ultrafilters for Boolean algebras respectively, which are easily shown by the facts $\mathrm{T}_1(x) = x$ and $\mathrm{T}_0(x) = x \to 0$ in the two-valued case.

Lemma 1. *Let P be an L-filter of an L-***VL***-algebra A. Then, P is a prime L-filter iff P is an ultra L-filter.*

Proof. Let P be a prime L-filter and $x \in A$. By $1 \in P$, $\bigvee\{\mathrm{T}_a(x)\,;\, a \in L\} \in P$. Thus, $\mathrm{T}_1(\bigvee\{\mathrm{T}_a(x)\,;\, a \in L\}) \in P$. Suppose $L = \{a_1, ..., a_n\}$. Since P is prime, there exist $b_1, ..., b_n \in L$ such that

$$b_1 \vee ... \vee b_n = 1 \text{ and } \mathrm{T}_{b_k}(\mathrm{T}_{a_k}(x)) \in P \text{ for } k = 1, ..., n.$$

By the axiom $\mathrm{T}_a(\mathrm{T}_b(x)) = 0$ (for $a \neq 0, 1$), all b_k's are equal to 0 or 1. Thus, by $b_1 \vee ... \vee b_n = 1$, some b_k is equal to 1, which implies $\mathrm{T}_1(\mathrm{T}_{a_k}(x)) \in P$. By the axiom $\mathrm{T}_1(x) \leq x$, we have $\mathrm{T}_{a_k}(x) \in P$. Hence, P is an ultra L-filter.

Let P be an ultra L-filter and $\mathrm{T}_c(x \vee y) \in P$. There is $a, b \in L$ such that

$$\mathrm{T}_a(x) \in P \text{ and } \mathrm{T}_b(y) \in P.$$

By $\mathrm{T}_a(x) \wedge \mathrm{T}_b(y) \leq \mathrm{T}_{a \vee b}(x \vee y)$, we have $\mathrm{T}_{a \vee b}(x \vee y) \in P$. Since $\mathrm{T}_c(x \vee y) \in P$ and since $\mathrm{T}_a(x) \wedge \mathrm{T}_b(x) = 0$ (for $a \neq b$), we can conclude that $a \vee b = c$. Hence, P is a prime L-filter.

Lemma 2. *Let P be an L-filter of an L-**VL**-algebra A. Then, P is a maximal L-filter iff P is an ultra L-filter.*

Proof. Let P be a maximal L-filter and $x \in A$. Suppose for contradiction that $\mathrm{T}_a(x) \notin P$ for all $a \in L$. Fix $a \in L$. Since P is maximal, there exists a term $\varphi \in A$ such that

$$\varphi = 0 \text{ and } \varphi \text{ is constructed from } \wedge, \mathrm{T}_1, \mathrm{T}_a(x) \text{ and the elements of } P.$$

Let $\psi = \mathrm{T}_1(\varphi)$. Note that $\psi = 0$. From $\mathrm{T}_1(\mathrm{T}_1(x)) = \mathrm{T}_1(x)$ and $\mathrm{T}_1(x \wedge y) = \mathrm{T}_1(x) \wedge \mathrm{T}_1(y)$, it follows that

$$\psi = \mathrm{T}_1(r_a \wedge \mathrm{T}_a(x)) \text{ for some } r_a \in P.$$

Hence, for all $a \in L$ there exists $r_a \in P$ such that

$$\mathrm{T}_1(r_a \wedge \mathrm{T}_a(x)) = 0.$$

From $\mathrm{T}_1(\mathrm{T}_a(x)) = \mathrm{T}_a(x)$, it follows that

$$\mathrm{T}_1(r_a \wedge \mathrm{T}_a(x)) = \mathrm{T}_1(r_a) \wedge \mathrm{T}_a(x) = 0 \text{ for all } a \in L.$$

Therefore, we have

$$(\bigwedge\{\mathrm{T}_1(r_a)\ ;\ a \in L\}) \wedge \mathrm{T}_a(x) = 0,$$

whence the following holds:

$$(\bigwedge\{\mathrm{T}_1(r_a)\ ;\ a \in L\}) \wedge (\bigvee\{\mathrm{T}_a(x)\ ;\ a \in L\}) = 0.$$

Thus, we have $\bigwedge\{\mathrm{T}_1(r_a)\ ;\ a \in L\} = 0$. Since $r_a \in P$, $\bigwedge\{\mathrm{T}_1(r_a)\ ;\ a \in L\} \in P$. Thus, $0 \in P$, which is a contradiction. Hence, P is an ultra L-filter.

Let P be an ultra L-filter and F an L-filter with $P \subset F$. Assume $x \in F$. Then, $\mathrm{T}_1(x) \in F$. Since there exists $a \in L$ with $\mathrm{T}_a(x) \in P$, we have $\mathrm{T}_a(x) \in F$. Thus, we have $\mathrm{T}_1(x) \wedge \mathrm{T}_a(x) \in F$. If $a \neq 1$, then $0 = \mathrm{T}_1(x) \wedge \mathrm{T}_a(x) \in F$. Hence, $a = 1$ and we have $\mathrm{T}_1(x) \in P$, which implies $x \in P$. Therefore, P is a maximal L-filter.

We obtain the next proposition from Lemma 1 and Lemma 2.

Proposition 3. *The following are equivalent.*

(i) P is a prime L-filter.
(ii) P is an ultra L-filter.
(iii) P is a maximal L-filter.

We can show an L-valued version of prime filter theorem for Boolean algebras. If L is the two-element Boolean algebra, the following theorem coincide with prime filter theorem for Boolean algebras.

Theorem 1. *Let A be an L-**VL**-algebra. Assume $x \neq y$ for $x, y \in A$. Then, there exist $a \in L$ and a prime L-filter P of A such that $\mathrm{T}_a(x) \in P$ and $\mathrm{T}_a(y) \notin P$.*

Proof. By the assumption, $x \leftrightarrow y \neq 1$. By the axiom "$\bigwedge_{a \in L}(\mathrm{T}_a(x) \leftrightarrow \mathrm{T}_a(y)) \leq x \leftrightarrow y$", we have $\bigwedge_{a \in L}(\mathrm{T}_a(x) \leftrightarrow \mathrm{T}_a(y)) \neq 1$, which implies that there is $a \in L$ such that $\mathrm{T}_a(x) \leftrightarrow \mathrm{T}_a(y) \neq 1$, i.e., $\mathrm{T}_a(x) \neq \mathrm{T}_a(y)$. We may assume that $\neg(\mathrm{T}_a(x) \leq \mathrm{T}_a(y))$. Let

$$F_0 = \{z \in A\,;\ \mathrm{T}_a(x) \leq z\}.$$

Note that $\mathrm{T}_a(y) \notin F_0$. Since $\mathrm{T}_1(\mathrm{T}_a(x)) = \mathrm{T}_a(x)$ and since T_1 is order-preserving, F_0 is an L-filter. Define X as the set of L-filters F such that $\mathrm{T}_a(x) \in F$ and $\mathrm{T}_a(y) \notin F$. Since $F_0 \in X$, X is not empty. It is easily shown that every chain of X has an upper bound. Thus, by Zorn's lemma, X contains a maximal element P. Clearly, $\mathrm{T}_a(x) \in P$ and $\mathrm{T}_a(y) \notin P$.

We claim that P is an ultra L-filter. If not, there is $z \in A$ such that $\forall c \in L$ $\mathrm{T}_c(z) \notin P$. Fix $c \in L$. Since P is maximal, there is a term $\varphi \in A$ such that

$$\varphi \leq \mathrm{T}_a(y) \text{ and } \varphi \text{ is constructed from } \wedge, \mathrm{T}_1, \mathrm{T}_c(z) \text{ and the elements of } P.$$

Let ψ be $\mathrm{T}_1(\varphi)$. Then, $\psi \leq \mathrm{T}_1(\mathrm{T}_a(y)) = \mathrm{T}_a(y)$, which leads to a contradiction by arguing as in the first paragraph of the proof of Lemma 2. Thus P is an ultra L-filter, whence P is a prime L-filter.

We can construct an L-valuation v_P from a prime L-filter P as follows.

Definition 6. *Let P be a prime L-filter of an L-**VL**-algebra A. Then, we define $v_P : A \to L$ by*

$$v_P(x) = a \Leftrightarrow \mathrm{T}_a(x) \in P.$$

We show that v_P is well-defined. Let $x \in A$. Since P is prime and since the L-valued excluded middle holds, there exists $a \in L$ with $\mathrm{T}_a(x) \in P$. If $a \neq b$ then $\mathrm{T}_a(x) \wedge \mathrm{T}_b(x) = 0$. Hence, if $\mathrm{T}_a(x) \in P$ and $\mathrm{T}_b(x) \in P$, then $a = b$.

Proposition 4. *Let P be a prime L-filter of an L-**VL**-algebra A. Then, v_P is a homomorphism from A to L.*

Proof. By $\mathrm{T}_1(1) = 1$, $v_P(1) = 1$. By $\mathrm{T}_0(0) = 1$, $v_P(0) = 0$.

We show that $v_P(\mathrm{T}_a(x)) = \mathrm{T}_a(v_P(x))$. Let b be $v_P(x)$. Then, $\mathrm{T}_b(x) \in P$. By the axiom $\mathrm{T}_b(x) \leq \mathrm{T}_{\mathrm{T}_a(b)}(\mathrm{T}_a(x))$, we have $\mathrm{T}_{\mathrm{T}_a(b)}(\mathrm{T}_a(x)) \in P$, which implies $v_P(\mathrm{T}_a(x)) = \mathrm{T}_a(v_P(x))$.

Next, we show that $v_P(x \to y) = v_P(x) \to v_P(y)$. Let a and b be $v_P(x)$ and $v_P(y)$ respectively. Then, $\mathrm{T}_a(x) \in P$ and $\mathrm{T}_b(y) \in P$. Thus, $\mathrm{T}_a(x) \wedge \mathrm{T}_b(y) \in P$. Since $\mathrm{T}_a(x) \wedge \mathrm{T}_b(y) \leq \mathrm{T}_{a \to b}(x \to y)$, we have $\mathrm{T}_{a \to b}(x \to y) \in P$, which implies $v_P(x \to y) = v_P(x) \to v_P(y)$.

In similar ways, we can prove that $v_P(x \wedge y) = v_P(x) \wedge v_P(y)$ and that $v_P(x \vee y) = v_P(x) \vee v_P(y)$.

Definition 7. *Let A be an L-**VL**-algebra.* $\mathrm{Spec}_L(A)$ *is defined as the set of all prime L-filters of A.*

As a Boolean algebra is embedded into a powerset algebra (Stone's representation theorem), an L-**VL**-algebra can be embedded into an L-valued powerset algebra L^S for a set S, where L^S is the set of all functions from S to L and the operations of L^S are defined pointwise (for example, $f \wedge g$ is defined by $(f \wedge g)(x) = f(x) \wedge g(x)$ for $f, g \in L^S$ and $x \in S$):

Theorem 2. *Let A be an L-**VL**-algebra and $S = \mathrm{Spec}_L(A)$. Define $\Phi : A \to L^S$ by*

$$\Phi(x) = (v_P(x))_{P \in S}.$$

Then, Φ is an embedding, i.e., an injective homomorphism.

Proof. By Proposition 4, v_P is a homomorphism. Since the operations of L^S are defined pointwise, Φ is a homomorphism. Let $x, y \in A$ with $x \neq y$. By Theorem 1, there exists a prime L-filter P such that $\mathrm{T}_a(x) \in P$ and $\mathrm{T}_a(y) \notin P$ for some $a \in L$. Thus, $v_P(x) \neq v_P(y)$, which implies $\Phi(x) \neq \Phi(y)$. Hence, Φ is injective.

Theorem 3. $\mathcal{V}(L)$ *coincides with the class of all L-**VL**-algebras. Moreover,* $\mathcal{Q}(L)$ *coincides with the class of all L-**VL**-algebras. Hence $\mathcal{V}(L) = \mathcal{Q}(L)$.*

Proof. Let A be an L-**VL**-algebra. Then, by Theorem 2, we have $A \in \mathcal{Q}(L)$ and therefore $A \in \mathcal{V}(L)$ by $\mathcal{Q}(L) \subset \mathcal{V}(L)$.

Conversely, let $A \in \mathcal{V}(L)$. Since the validity of equations is preserved under taking homomorphic images, subalgebras and products and since L satisfies the axioms in Definition 3, A also satisfies the axioms and therefore A is an L-**VL**-algebra. Thus, by $\mathcal{Q}(L) \subset \mathcal{V}(L)$, if $A \in \mathcal{Q}(L)$ then A is an L-**VL**-algebra. This completes the proof.

Proposition 5. *Define $f : \mathrm{Spec}_L(A) \to \mathrm{Hom}(A, L)$ by*

$$f(P) = v_P.$$

Then, f is a bijection.

Proof. By Proposition 4, v_P is a homomorphism. It is easy to show that f is injective. To show that f is surjective, assume $v \in \mathrm{Hom}(A, L)$. Clearly, $v^{-1}(\{1\})$ is a prime L-filter. Let P be $v^{-1}(\{1\})$. We claim that $f(P) = v$, i.e., $v_P(x) = v(x)$ for all $x \in A$. Since $v(\mathrm{T}_{v(x)}(x)) = \mathrm{T}_{v(x)}(v(x)) = 1$, we have $\mathrm{T}_{v(x)}(x) \in v^{-1}(\{1\})$, whence $v_P(x) = v(x)$.

Thus, we can see $\mathrm{Spec}_L(A)$ as the set of all L-valuations on A.

Remark 1. By using T_a's, it is easily verified that L forms a semi-primal algebra, where note that if L is additionally endowed with all truth constants then L forms a primal algebra (for the definitions, see [3]). Thus, by Theorem 3, Proposition 5 and [3, Theorem 3.3.14], we can obtain a strong duality for L-**VL**-algebras, which implies that, for an L-**VL**-algebra A, the embedding $\Phi : A \to L^S$ gives a Boolean product representation of A (for the definition, see [2]), where $\mathrm{Spec}_L(A)$ is equipped with the topology generated by $\{v \in \mathrm{Spec}_L(A); v(a) = 1\}$ for $a \in A$.

By Theorem 2 and Theorem 3, we obtain the following theorem, which contains the completeness with respect to L-valued semantics.

Theorem 4. *Let $x, y \in \mathbf{Form}$. The following are equivalent:*

(i) $x = y$ holds in any L-$\mathbf{VL}$-algebras;
(ii) $v(x) = v(y)$ for any L-valuation v;
(iii) $x = y$ holds in L.

By Theorem 4, it is decidable whether $x = y$ holds in any L-$\mathbf{VL}$-algebras.

Finally, we remark that the logic determined by L-$\mathbf{VL}$-algebras (i.e., the free L-$\mathbf{VL}$-algebra generated by the propositional variables) is not strictly the same as Fitting's L-valued logic, but they are closely related through completeness.

2.2 Basic Properties of U_a and D_a

We define unary connectives U_a and D_a as follows.

Definition 8. *For $a \in L$, we define U_a and D_a by:*

$$\mathrm{U}_a(x) = \bigvee\{\mathrm{T}_b(x) \,;\, a \le b\};$$
$$\mathrm{D}_a(x) = \bigvee\{\mathrm{T}_b(x) \,;\, a \ge b\}.$$

$\mathrm{U}_a(x)$ (resp. $\mathrm{D}_a(x)$) intuitively states that the truth value of x is more than (resp. less than) or equal to a. Actually, we can easily verify the next proposition.

Proposition 6. *Let v be an L-valuation. Then, the following hold:*

(i) $\mathrm{U}_a(b) = 1$ (if $a \le b$) and $\mathrm{U}_a(b) = 0$ (otherwise);
(ii) $v(\mathrm{U}_a(x)) = 1$ (if $a \le v(x)$) and $v(\mathrm{U}_a(x)) = 0$ (otherwise);
(iii) $\mathrm{D}_a(b) = 1$ (if $a \ge b$) and $\mathrm{D}_a(b) = 0$ (otherwise);
(iv) $v(\mathrm{D}_a(x)) = 1$ (if $a \ge v(x)$) and $v(\mathrm{D}_a(x)) = 0$ (otherwise).

The next proposition shows that U_a's are inter-definable with T_a's and that D_a's are inter-definable with T_a's.

Proposition 7. *Let $a \in L$. Then, the following hold:*

(i) $\mathrm{T}_a(x) = \mathrm{U}_a(x) \wedge (\bigwedge\{\mathrm{U}_b(x) \to 0 \,;\, a < b\})$ holds in any L-$\mathbf{VL}$-algebras;
(ii) $\mathrm{T}_a(x) = \mathrm{D}_a(x) \wedge (\bigwedge\{\mathrm{D}_b(x) \to 0 \,;\, a > b\})$ holds in any L-$\mathbf{VL}$-algebras.

Proof. We show (i). By Theorem 4, it suffices to show that the above equation holds in L. Assume $x \in L$. If $x = a$, then $\mathrm{U}_a(x) = 1$ and $\mathrm{U}_b(x) \to 0 = 1$ for $b \in L$ with $a < b$, whence the equation holds. If $x > a$, then $\bigwedge\{\mathrm{U}_b(x) \to 0 \,;\, a < b\} = 0$ by $\mathrm{U}_x(x) \to 0 = 0$, whence the equation holds. If $x \not\ge a$, then $\mathrm{U}_a(x) = 0$, whence the equation holds. (ii) is proved in a similar way.

Remark 2. By replacing T_a(-) with $\mathrm{U}_a(\text{-}) \wedge (\bigwedge\{\mathrm{U}_b(\text{-}) \to 0 \,;\, a < b\})$ in Definition 3, we can obtain an algebraic axiomatization of L-$\mathbf{VL}$ using U_a's as primitives

instead of T_a's. In a similar way, we can obtain an axiomatization using D_a's as primitives instead of T_a's.

It seems that our axiomatization using T_a's is simpler than that using U_a's (or D_a's). For this reason, we first axiomatized $\boldsymbol{L}$-**VL** using T_a's rather than U_a's (or D_a's). However, using U_a's (or D_a's) also has some advantages (see Section 4 below).

In the remaining part of this paper, we do not consider D_a's, and consider only U_a's, though similar results as for U_a's hold also for D_a's.

Proposition 8. *The following equations hold in any L-**VL**-algebra:*

(i) $\mathrm{U}_1(x) = \mathrm{T}_1(x)$, $\mathrm{T}_a(x) \leq \mathrm{U}_a(x)$*;*
(ii) $\mathrm{U}_a(0) = 0$ *(for $a \neq 0$),* $\mathrm{U}_0(x) = 1$, $\mathrm{U}_a(1) = 1$*;*
(iii) $\mathrm{U}_b(\mathrm{U}_a(x)) = \mathrm{U}_a(x)$ *(for $b \neq 0$),* $\mathrm{U}_a(x) \leq \mathrm{U}_b(x)$ *(for $b \leq a$);*
(iv) $\mathrm{U}_a(x \wedge y) = \mathrm{U}_a(x) \wedge \mathrm{U}_a(y)$*.*

Proof. We prove only (iv), since the others can be shown in similar ways. By Theorem 4, it suffices to show that $\mathrm{U}_a(x \wedge y) = \mathrm{U}_a(x) \wedge \mathrm{U}_a(y)$ holds in L. Assume $x, y \in L$. If $\mathrm{U}_a(x \wedge y) = 1$, then $a \leq x \wedge y$, which implies $a \leq x$ and $a \leq y$. If $\mathrm{U}_a(x \wedge y) = 0$, then $a \nleq x \wedge y$, which implies either $a \nleq x$ or $a \nleq y$. Thus, $\mathrm{U}_a(x \wedge y) = \mathrm{U}_a(x) \wedge \mathrm{U}_a(y)$.

Note that U_a is not distributive over $\vee$ in general, though $\mathrm{U}_a(x) \vee \mathrm{U}_a(y) \leq \mathrm{U}_a(x \vee y)$ holds in general. If L is totally ordered, U_a is distributive over $\vee$.

We can give a different description of v_P by exploiting U_a.

Proposition 9. *Let P be a prime L-filter of an L-**VL**-algebra A. Then, the following holds: $v_P(x) = \bigvee\{a\,;\ \mathrm{U}_a(x) \in P\}$ for $x \in A$.*

Proof. It suffices to show that $\mathrm{U}_a(x) \in P$ is equivalent to $a \leq v_P(x)$. Now, $a \leq v_P(x)$ is equivalent to $\exists b \geq a\ \mathrm{T}_b(x) \in P$, which is equivalent to $\mathrm{U}_a(x) \in P$ by $\mathrm{T}_b(x) \leq \mathrm{U}_b(x)$ and $\mathrm{U}_b(x) \leq \mathrm{U}_a(x)$ for $a \leq b$.

3 Lattice-Valued Modal Logic $\boldsymbol{L}$-ML

We define L-valued modal logic $\boldsymbol{L}$-**ML** from a semantical point of view. The connectives of $\boldsymbol{L}$-**ML** are a unary connective $\Box$ and the connectives of $\boldsymbol{L}$-**VL**. **PV** denotes the set of propositional variables. Then, the formulas of $\boldsymbol{L}$-**ML** are recursively defined in a usual way. $\mathbf{Form}_\Box$ denotes the set of formulas of $\boldsymbol{L}$-**ML**.

Definition 9. *Let (M, R) be a Kripke frame. Then, v is a Kripke L-valuation on (M, R) iff v is a function from $M \times \mathbf{Form}_\Box$ to L and satisfies the following properties for each $w \in M$:*

$$\begin{aligned} v(w, \Box x) &= \bigwedge\{v(w', x)\,;\ wRw'\}; \\ v(w, \mathrm{T}_a(x)) &= \mathrm{T}_a(v(w, x)); \\ v(w, x \wedge y) &= \inf(v(w, x), v(w, y)); \\ v(w, x \vee y) &= \sup(v(w, x), v(w, y)); \end{aligned}$$

$$v(w, x \to y) = v(w, x) \to_L v(w, y);$$
$$v(a) = a \quad \text{for } a = 0, 1.$$

Then, we call (M, R, v) an L-valued (modal) Kripke model.

A formula x is called a valid formula of L-**ML** iff $v(w, x) = 1$ in any L-valued Kripke model (M, R, v) and any $w \in M$. If L is the two-element Boolean algebra, the valid formulas of L-**ML** coincide with those of classical modal logic.

3.1 An Algebraic Axiomatization of L-ML

We give an algebraic axiomatization of L-**ML** as follows.

Definition 10. $(A, \wedge, \vee, \to, \mathrm{T}_a(a \in L), \Box, 0, 1)$ *is an L-**ML**-algebra iff it satisfies the following axioms:*

(i) $(A, \wedge, \vee, \to, \mathrm{T}_a(a \in L), 0, 1)$ *forms an L-**VL**-algebra;*
(ii) $\Box(x \wedge y) = \Box x \wedge \Box y$, $\Box 1 = 1$*;*
(iii) $\Box \mathrm{U}_a(x) = \mathrm{U}_a(\Box x)$ *(for all $a \in L$).*

To show the completeness, we define the notion of an L-valued canonical model, which is a generalization of a canonical model of (classical) modal logic.

Definition 11. *Let A be an L-**ML**-algebra. For $p \in \mathrm{Spec}_L(A)$ and $a \in L$, let*

$$p_a = \{\mathrm{U}_a(x)\, ;\ \mathrm{U}_a(\Box x) \in p\}.$$

Define a binary relation $R_\Box$ on $\mathrm{Spec}_L(A)$ by

$$pR_\Box q \Leftrightarrow \forall a \in L\; p_a \subset q.$$

*Define $v : \mathrm{Spec}_L(A) \times$ **Form** $\to L$ by*

$$v(p, x) = v_p(x),$$

where v_p is defined in Definition 6. Then, $(\mathrm{Spec}_L(A), R_\Box, v)$ is called the L-valued canonical model of A.

Lemma 3. *Under the notation of Definition 11, p_a is closed under $\wedge$.*

Proof. Assume that $\mathrm{U}_a(x) \in p_a$ and $\mathrm{U}_a(y) \in p_a$. Then, it is easy to verify that $\mathrm{U}_a(\Box x) \in p$ and $\mathrm{U}_a(\Box y) \in p$. Thus, $\mathrm{U}_a(\Box x) \wedge \mathrm{U}_a(\Box y) \in p$. From $\mathrm{U}_a(\Box x) \wedge \mathrm{U}_a(\Box y) = \mathrm{U}_a(\Box(x \wedge y))$, it follows that $\mathrm{U}_a(x \wedge y) \in p_a$. Since $\mathrm{U}_a(x \wedge y) = \mathrm{U}_a(x) \wedge \mathrm{U}_a(y)$, we can conclude that p_a is closed under $\wedge$.

The next lemma is the most important part of our completeness proof.

Lemma 4. *Let A be an L-**ML**-algebra, $p \in \mathrm{Spec}_L(A)$ and $a \in L$. Then, for $x \in A$, the following are equivalent:*

(i) $\mathrm{U}_a(\Box x) \in p$*;*
(ii) $\forall q \in \mathrm{Spec}_L(A)$ *($pR_\Box q$ implies $\mathrm{U}_a(x) \in q$).*

Proof. We show that (ii) implies (i). To prove the contrapositive, assume that $\mathrm{U}_a(\Box x) \notin p$. We show that there exists $q \in \mathrm{Spec}_L(A)$ such that $pR_\Box q$ and $\mathrm{U}_a(x) \notin q$. Let F be the L-filter generated by $\bigcup\{p_b \,;\, b \in L\}$.

We claim that $\mathrm{U}_a(x) \notin F$. Suppose for contradiction that $\mathrm{U}_a(x) \in F$. Then, there exists $\varphi \in A$ such that $\varphi \leq \mathrm{U}_a(x)$ and φ is constructed from $\wedge$, T_1 and the elements of $\bigcup\{p_b \,;\, b \in L\}$. Since $\mathrm{T}_1(\mathrm{U}_b(x)) = \mathrm{U}_b(x)$ holds in general and since p_b is closed under $\wedge$ by Lemma 3.1, we may assume that

$$\varphi = \bigwedge\{\mathrm{U}_b(x_b) \,;\, b \in L\},$$

where $\mathrm{U}_b(x_b)$ is an element of p_b for each $b \in L$. By $\varphi \leq \mathrm{U}_a(x)$, we have $\Box\varphi \leq \Box\mathrm{U}_a(x)$. Now,

$$\Box\varphi = \bigwedge\{\mathrm{U}_b(\Box x_b) \,;\, b \in L\}.$$

Since $\mathrm{U}_b(\Box x_b) \in p$, we have $\Box\mathrm{U}_a(x) \in p$, which is a contradiction. Thus, we can conclude that $\mathrm{U}_a(x) \notin F$.

Let X be the set of L-filters G such that $\mathrm{U}_a(x) \notin G$ and $F \subset G$. Note that $F \in X$. By Zorn's lemma, X has a maximal element q. Arguing as in the second paragraph of the proof of Theorem 1, we can show that q is an ultra L-filter, whence $q \in \mathrm{Spec}_L(A)$. Clearly, $\mathrm{U}_a(x) \notin q$. By $p_b \subset F$ for each b, we have $pR_\Box q$.

It is straightforward to show that (i) implies (ii).

By the above lemma, we can show that an L-valued canonical model is actually an L-valued Kripke model:

Proposition 10. *Let A be an L-**ML**-algebra. Then, the L-valued canonical model $(\mathrm{Spec}_L(A), R_\Box, v)$ is an L-valued Kripke model.*

Proof. We show that v is a Kripke L-valuation on $(\mathrm{Spec}_L(A), R_\Box)$. Fix $p \in \mathrm{Spec}_L(A)$. By Proposition 4, v_p preserves the conncectives of L-**VL**. Thus, it suffices to show that $v_p(\Box x) = \bigwedge\{v_q(x) \,;\, pR_\Box q\}$. Let c ($\in L$) be such that $\mathrm{T}_c(\Box x) \in p$ (i.e., $v_p(\Box x) = c$). Since $\mathrm{U}_c(\Box x) \in p$, it follows from Lemma 4 that $c \leq \bigwedge\{v_q(x) \,;\, pR_\Box q\}$. If $c < a$, then $\mathrm{U}_a(\Box x) \notin p$ and therefore, by Lemma 4, $\bigwedge\{v_q(x) \,;\, pR_\Box q\} < a$. Hence, we have $v_p(\Box x) = c = \bigwedge\{v_q(x) \,;\, pR_\Box q\}$.

By using the above facts, we obtain the following completeness theorem.

Theorem 5. *Let $x, y \in$ **Form**$_\Box$. Then, the following are equivalent:*

*(i) $x = y$ holds in any L-**ML**-algebras;*
(ii) $v(w, x) = v(w, y)$ in any L-valued Kripke model (M, R, v) and any $w \in M$.

Proof. By straightforward computation, we can verify that (i) implies (ii). We show that (ii) implies (i). To prove the contrapositive, assume that $x \neq y$ in some L-**ML**-algebra A. Consider the L-valued canonical model $(\mathrm{Spec}_L(A), R_\Box, v)$, which is an L-valued Kripke model by Proposition 10. By Theorem 2, there exists $p \in \mathrm{Spec}_L(A)$ such that $v_p(x) \neq v_p(y)$, i.e., $v(p, x) \neq v(p, y)$.

3.2 Finite Model Property

By an L-valued version of the well known filtration method, we can prove the finite model property of L-**ML**.

Theorem 6. *For $x, y \in \mathbf{Form}_\Box$, the following are equivalent:*

(i) $x = y$ holds in any L-$\mathbf{ML}$-algebras;
(ii) $v(w, x) = v(w, y)$ in any finite L-valued Kripke model (M, R, v) and any $w \in M$.

Proof. It is enough to show that (ii) implies (i). To prove the contrapositive, assume that $x \neq y$ in some L-**ML**-algebra A. Consider $(\mathrm{Spec}_L(A), R_\Box, v)$. Let X (resp. Y) be the set of all subformulas of x (resp. y). Define an equivalence relation $\sim$ on $\mathrm{Spec}_L(A)$ by

$$p \sim q \Leftrightarrow \forall z \in X \cup Y \; v(p, z) = v(q, z).$$

Since L is finite and since X and Y are finite, $\mathrm{Spec}_L(A)/\sim$ is a finite set. We denote by $[p]$ the set $\{q \in \mathrm{Spec}_L(A) \; ; \; p \sim q\}$. Let

$$p'_a = \{\mathrm{U}_a(z) \; ; \; \mathrm{U}_a(\Box z) \in p \text{ and } \Box z \in X \cup Y\}.$$

Define a binary relation S on $\mathrm{Spec}_L(A)/\sim$ by

$$[p]S[q] \Leftrightarrow \forall a \in L \; p'_a \subset q.$$

We can consider a Kripke L-valuation v' on $(\mathrm{Spec}_L(A)/\sim, S)$ such that $v'([p], z) = v(p, z)$ for all $z \in \mathbf{PV} \cap (X \cup Y)$ and all $p \in \mathrm{Spec}_L(A)$. By induction on the formulas $X \cup Y$, we show the fact that

$$v'([p], z) = v(p, z) \text{ for all } z \in X \cup Y \text{ and all } p \in \mathrm{Spec}_L(A).$$

We show only the case that z has the form $\Box z'$, since the other cases are easily verfied. It suffices to show that, for any $a \in L$,

$$v'([p], \Box z') \geq a \text{ iff } v(p, \Box z') \geq a.$$

If $v'([p], \Box z') \geq a$, then, by the induction hypothesis, we have $\bigwedge\{v(q, z'); [p]S[q]\} \geq a$ and therefore $\bigwedge\{v(q, z') \; ; \; pR_\Box q\} \geq a$, since $pR_\Box q$ implies $[p]S[q]$. If $v(p, \Box z') \geq a$, then $v(q, z') \geq a$ for any q with $[p]S[q]$ by the definition of S and therefore $v'([q], z') \geq a$ for any $[q]$ with $[p]S[q]$.

By Theorem 2, there exists $p \in \mathrm{Spec}_L(A)$ such that $v(p, x) \neq v(p, y)$, whence we have $v'([p], x) \neq v'([p], y)$ by the above fact.

By Theorem 6, it is decidable whether $x = y$ holds in any L-**ML**-algebras.

3.3 *L*-Valued Modal Logic of S4 Type

L-valued **S4**-type modal logic L-**S4** is naturally defined as follows.

Definition 12. *An L-valued (modal) Kripke model (M, R, v) is an L-***S4** *Kripke model iff R is reflexive and transitive. Then, x is a valid formula of L-***S4** *iff $v(w, x) = 1$ in any L-***S4** *Kripke model (M, R, v) and any $w \in M$.*

We define the class of L-**S4**-algebras, which is actually the algebraic counterpart of L-**S4** by Theorem 7 below.

Definition 13. *An L-***S4***-algebra is defined as an L-***ML***-algebra with the additional axioms $\Box x \leq x$ and $\Box x \leq \Box\Box x$.*

Lemma 5. *Let A be an L-***S4***-algebra and $(\mathrm{Spec}_L(A), R_\Box, v)$ the L-valued canonical model. Then, the following holds: $pR_\Box q$ iff $\forall a \in L\ p_a \subset q_a$. Hence, $R_\Box$ is reflexive and transitive.*

Proof. Assume $pR_\Box q$. Let $a \in L$ and $\mathrm{U}_a(x) \in p_a$. Then, it is easily verified that $\mathrm{U}_a(\Box x) \in p$. Since $\mathrm{U}_a(\Box x) = \mathrm{U}_a(\Box\Box x)$, we have $\mathrm{U}_a(\Box x) \in p_a$, whence $\mathrm{U}_a(\Box x) \in q$ by the assumption and then we have $\mathrm{U}_a(x) \in q_a$

Assume that $\forall a \in L\ p_a \subset q_a$. Let $a \in L$ and $\mathrm{U}_a(x) \in p_a$. Then, by the assumption, $\mathrm{U}_a(x) \in q_a$ and therefore $\mathrm{U}_a(\Box x) \in q_a$ by $\Box x \leq x$, whence we have $\mathrm{U}_a(x) \in q_a$.

By the above lemma, we can prove the following completeness theorem.

Theorem 7. *Let $x, y \in$* $\mathbf{Form}_\Box$*. Then, the following are equivalent:*

*(i) $x = y$ holds in any L-***S4***-algebras;*
*(ii) $v(w, x) = v(w, y)$ in any L-***S4** *Kripke model (M, R, v) and any $w \in M$.*

Proof. It is straightforward to verify that (i) implies (ii). We show that (ii) implies (i). To prove the contrapositive, assume that $x \neq y$ in some L-**S4**-algebra A. Let $(\mathrm{Spec}_L(A), R_\Box, v)$ be the L-valued canonical model of A. Then, by Lemma 5, $(\mathrm{Spec}_L(A), R_\Box, v)$ is an L-**S4** Kripke model. The remaining part of the proof is the same as Theorem 5.

We define L-valued intuitionistic logic L-**IL** from a semantical point of view. The formulas of L-**IL** are the same as those of L-**VL**.

Definition 14. *Let (M, R) be a preorder set. Then, u is an intuitionistic Kripke L-valuation on (M, R) iff u is a function from $M \times$* **Form** *to L and satisfies the following properties for each $w \in M$:*

$$\begin{aligned} u(w, p) &\leq u(w', p) \ \ (\text{for } wRw'); \\ u(w, x \to y) &= \bigwedge \{u(w', x) \to_L u(w', y)\, ;\, wRw'\}; \\ u(w, \mathrm{T}_a(x)) &= \mathrm{T}_a(u(w, x)); \\ u(w, x \wedge y) &= \inf(u(w, x), u(w, y)); \\ u(w, x \vee y) &= \sup(u(w, x), u(w, y)); \\ u(a) &= a \quad \text{for } a = 0, 1. \end{aligned}$$

Then we call (M, R, u) an L-valued intuitionistic Kripke model.

Then x is a valid formula of L-**IL** iff $u(w, x) = 1$ in any L-valued intuitionistic Kripke model (M, R, u) and any $w \in M$. If L is the two-element Boolean algebra, the valid formulas of L-**IL** coincide with those of intuitionistic logic.

Definition 15. *Gödel translation G from L-**IL** to L-**S4** is the function from **Form** to $\mathbf{Form}_\Box$ satisfying the following properties:*

$$\begin{aligned} G(p) &= \Box p \quad \text{for } p \in \mathbf{PV}; \\ G(x \to y) &= \Box(G(x) \to G(y)); \\ G(\mathrm{T}_a(x)) &= \mathrm{T}_a(G(x)); \\ G(x \wedge y) &= G(x) \wedge G(y); \\ G(x \vee y) &= G(x) \vee G(y); \\ G(a) &= a \quad \text{for } a = 0, 1. \end{aligned}$$

A Gödel-Tarski-McKinsey style theorem holds between L-**IL** and L-**S4**:

Theorem 8. *Let $x \in$ **Form**. Then, the following are equivalent:*

*(i) x is a valid formula of L-**IL**;*
*(ii) $G(x)$ is a valid formula of L-**S4**.*

Proof. We show that (i) implies (ii). To prove the contrapositive, suppose that $v(w, G(x)) \neq 1$ in some L-**S4** Kripke model (M, R, v) and some $w \in M$. Consider the L-valued intuitionistic Kripke model (M, R, u) defined by $u(w, p) = v(w, \Box p)$ for $p \in \mathbf{PV}$. Since R is a preorder, wRw' implies $u(w, p) \leq u(w', p)$ for $p \in \mathbf{PV}$. By induction on formulas, we prove the fact that $u(w, x) = v(w, G(x))$ for all $x \in$ **Form**. If $p \in \mathbf{PV}$, then $u(w, p) = v(w, \Box p) = v(w, G(p))$. If x has the form $y \to z$, then we proceed as follows:

$$\begin{aligned} u(w, y \to z) &= \bigwedge \{u(w', y) \to u(w', z)\ ;\ wRw'\} \\ &= \bigwedge \{v(w', G(y)) \to v(w', G(z))\ ;\ wRw'\} \\ &= \bigwedge \{v(w', G(y) \to G(z))\ ;\ wRw'\} \\ &= v(w, \Box(G(y) \to G(z))). \end{aligned}$$

The other cases are easily verified. It follows from the above fact that $u(w, x) = v(w, G(x)) \neq 1$ in (M, R, u). Hence, x is not a valid formula of L-**IL**.

We show that (ii) implies (i). To prove the contrapositive, suppose that $u(w, x) \neq 1$ in some L-valued intuitionistic Kripke model (M, R, u) and some $w \in M$. Consider the L-**S4** Kripke model (M, R, v) defined by $v(w, p) = u(w, p)$ for $p \in \mathbf{PV}$. By induction on formulas, we prove that

$$v(w, G(x)) = u(w, x) \text{ for all } x \in \mathbf{Form}.$$

If $p \in \mathbf{PV}$, then we have $u(w, p) = \bigwedge\{u(w', p); wRw'\} = v(w, \Box p) = v(w, G(p))$, since R is a preorder and since if wRw' then $u(w, p) \leq u(w', p)$. The other cases are easy to check. Therefore, $v(w, G(x)) = u(w, x) \neq 1$ in (M, R, v). Hence, $G(x)$ is not a valid formula of L-**S4**.

The above proof holds even if L is an infinite complete Heyting algebra.

4 Future Work

The Jónsson-Tarski duality ([1]) is one of the most important results for (classical) modal logic. Our future work will be to obtain a Jónsson-Tarski style duality for L-**VL** and L-**ML**. We conjecture that we can apply the theory of canonical extensions ([7],[8],[9]) to obtain such a duality, since we have the following facts: (i) These logics can be axiomatized using U_a's instead of T_a's (2.2 Remark 2); (ii) U_a's preserve finite meets by Proposition 8, though T_a's do not in general; (iii) the class of L-**VL**-algebras axiomatized using U_a's is finitely generated by Theorem 3 and therefore is closed under canonical extensions by (ii) and [7, Corollary 6.9].

Acknowledgements. The author would like to thank the anonymous referees for their invaluable comments and suggestions, especially those on the future work and related facts. The author is also grateful to Prof. Takashi Sakuragawa, Kentaro Sato and Katsuhiko Sano for their helpful advices.

References

1. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic, CUP (2001)
2. Burris, S., Sankappanavar, H.P.: A course in universal algebra. Springer, Heidelberg (1981)
3. Clark, D.M., Davey, B.A.: Natural dualities for the working algebraist, CUP (1998)
4. Grätzer, G.A.: Universal algebra. Springer, Heidelberg (1979)
5. Eleftheriou, P.E., Koutras, C.D.: Frame constructions, truth invariance and validity preservation in many-valued modal logic. J. Appl. Non-Classical Logics 15, 367–388 (2005)
6. Epstein, G.: The lattice theory of Post algebras. Trans. Amer. Math. Soc. 95, 300–317 (1960)
7. Gehrke, M., Harding, J.: Bounded lattice expansions. J. Algebra 239, 345–371 (2001)
8. Gehrke, M., Jónsson, B.: Bounded distributive lattice expansions. Math. Scand 94, 13–45 (2004)
9. Gehrke, M., Nagahashi, H., Venema, Y.: A Sahlqvist theorem for distributive modal logic. Ann. Pure Appl. Logic 131, 65–102 (2005)
10. Fitting, M.C.: Many-valued modal logics. Fund. Inform. 15, 235–254 (1991)
11. Fitting, M.C.: Many-valued modal logics II. Fund. Inform. 17, 55–73 (1992)
12. Fitting, M.C.: Tableaus for many-valued modal logic. Studia Logica 55, 63–87 (1995)
13. Koutras, C.D., Nomikos, C., Peppas, P.: Canonicity and completeness results for many-valued modal logics. J. Appl. Non-Classical Logics 12, 7–41 (2002)
14. Koutras, C.D., Peppas, P.: Weaker axioms, more ranges. Fund. Inform. 51, 297–310 (2002)
15. Koutras, C.D.: A catalog of weak many-valued modal axioms and their corresponding frame classes. J. Appl. Non-Classical Logics 13, 47–72 (2003)

A General Setting for the Pointwise Investigation of Determinacy

Yurii Khomskii*

Institute of Logic, Language and Computation
University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
The Netherlands

Abstract. It is well-known that if we assume a large class of sets of reals to be determined then we may conclude that all sets in this class have certain regularity properties: we say that determinacy implies regularity properties *classwise*. In [Lö05] the *pointwise* relation between determinacy and certain regularity properties (namely the Marczewski-Burstin algebra of arboreal forcing notions and a corresponding weak version) was examined.

An open question was how this result extends to topological forcing notions whose natural measurability algebra is the class of sets having the Baire property. We study the relationship between the two cases, and using a definition which adequately generalizes both the Marczewski-Burstin algebra of measurability and the Baire property, prove results similar to [Lö05].

We also show how this can be further generalized for the purpose of comparing algebras of measurability of various forcing notions.

1 Introduction

The classical theorems due to Mycielski-Swierczkowski, Banach-Mazur and Morton Davis respectively state that under the Axiom of Determinacy all sets of reals are Lebesgue measurable, have the Baire property and the perfect set property (see, e.g., [Ka94, pp 373–377]). In fact, these proofs give *classwise* implications, i.e., if $\boldsymbol{\Gamma}$ is a boldface pointclass (closed under continuous preimages and intersections with basic open sets) such that all sets in $\boldsymbol{\Gamma}$ are determined, then all sets in $\boldsymbol{\Gamma}$ have the corresponding regularity property. The proofs do *not*, however, show that from the assumption "A is determined" one can conclude "A is regular", i.e., they do not give us *pointwise* implications. So a natural question is: what is the strength of the statement "A is determined", and which properties of A follow from that statement?

That the strength of determinacy is in classwise rather than pointwise consequences is not unexpected—after all, it is easy to construct sets that are determined for trivial reasons. Still, if the regularity properties themselves are "weak"

* The author is supported from a Mosaic grant (no 017.004.066) by the Netherlands Organisation for Scientific Research (NWO).

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 185–195, 2009.

in a certain sense, the relationship is not as clear. In [Lö05], where the pointwise relationship between determinacy and Marczewski-Burstin measurability algebras (connected to so-called arboreal forcing notions) was first analyzed, we indeed have the expected result for the algebras themselves but more interesting ones if we consider their "weak" or "local" counterparts.

This investigation started with the question whether the results of [Lö05] can be transferred to the more difficult scenario of topological forcing notions whose natural algebra of measurability is not the Marczewski-Burstin algebra but the Baire property in the corresponding topology. In the process of studying this question, however, certain basic properties of arboreal forcings and their measurability algebras came to light, which forced us to adapt the definitions as well as the actual question. Partly motivated by recent work of Daisuke Ikegami [Ik08], we are adapting a different definition of arboreal forcings, and giving a new definition of a measurability notion. Using these new definitions we are able to generalize and improve [Lö05] while covering both the non-topological and the new topological cases. Our two main results here are Theorem 4.3 and Theorem 5.5.

In the last section we also show how the methods can be generalized for the purpose of comparing algebras of measurability of various forcing notions.

We should note that Definition 2.2 below gives far less freedom than [Lö05, Section 2.1], but there are good reasons for adopting it: firstly, Fact 2.3 could not be proved without it, secondly, one would be able to construct some very simple sets (e.g., closed in the standard topology) that are non-measurable. In short, the new definition eliminates "pathological cases" and makes sure that our forcing notions are somewhat reasonable. This, of course, also eliminates most of the crucial examples considered in [Lö05, Sections 5 and 6]. As a result, our conclusions differ from [Lö05] on certain points, but we feel that the new analysis is more intuitively satisfying and has more practical relevance because it is immune to artificial counterexamples.

2 Definitions and Preliminaries

We start by fixing some simple concepts about descriptive-theoretic trees:

Definition 2.1. *Let $T \subseteq \omega^{<\omega}$ or $2^{<\omega}$ be a tree.*

1. *For $t \in T$ we write $\mathrm{Succ}_T(t) := \{s \in T \mid \exists n(s = t^\frown\langle n\rangle)\}$ to denote the set of immediate successors of t.*
2. *A node $t \in T$ is called*
   - splitting *if $|\mathrm{Succ}_T(t)| > 1$ and* non-splitting *otherwise.*
   - ω-splitting *if $|\mathrm{Succ}_T(t)| = \omega$ and* n-splitting *if $|Succ_T(t)| = n < \omega$.*
   - totally splitting *if $\forall n\ (t^\frown\langle n\rangle \in T)$.*
3. *The* stem of T, *notation* $\mathrm{stem}(T)$, *is the largest $s \in T$ such that all $t \subsetneq s$ are non-splitting.*
4. *$[T]$ denotes the set of branches through T, i.e., $\{x \mid \forall n\ (x{\restriction}n \in T)\}$.*

Although we try to keep the notions T and $[T]$ separated, occasionally we will use the two objects interchangeably, since it makes arguments simpler and can cause no harm. Finally, we note that all trees considered in this paper are assumed to be *pruned*, i.e., every node has at least one successor.

We are ready to define arboreal forcing notions:

Definition 2.2. *A forcing partial order* $(\mathbb{P}, \leq)$ *is called* arboreal *if it is a collection of perfect trees on* ω^ω *(or* 2^ω*), ordered by inclusion, with the extra condition that*

$$\forall P \in \mathbb{P}\, \forall t \in P\, \exists Q \leq P\ (t \subseteq \text{stem}(Q)).$$

It is called topological *if the set of conditions* $\{[P] \mid P \in \mathbb{P}\}$ *forms a topology base for some topology on the set* ω^ω *(resp.* 2^ω*), and* non-topological *otherwise.*

Examples of standard non-topological arboreal forcings include Sacks forcing, Miller forcing, Laver forcing, Silver forcing and many more (for a definition see e.g. [BaJu95, Je86].) Examples of standard topological forcings are Cohen forcing, Hechler forcing, eventually different forcing and Matthias forcing. Cohen forcing generates the standard topology, while Hechler and eventually different forcing generate the *dominating topology* and the *eventually different topology*, respectively. Matthias forcing generates the *Ellentuck topology* (due to Erik Ellentuck [El74]).

The following fact is a straightforward consequence of our definition.

Lemma 2.3. *If* $\mathbb{P}$ *is an arboreal forcing notion, then* $\mathbb{P}$ *is separative. Moreover, we have for all* $P, Q \in \mathbb{P}$*, if* $P \not\leq Q$ *then* $\exists R \leq P$ *s.t.* $[R] \cap [Q] = \varnothing$ *(we say* $\mathbb{P}$ *is* strongly separative.*)*

Proof. Suppose $P \not\leq Q$. Then there is $t \in P \setminus Q$, so by definition there must be an $R \leq P$ with $t \subseteq \text{stem}(R)$. But then $[R] \cap [Q] = \varnothing$. □

Since this paper is about consequences of determinacy, let us also give that definition. There are a number of equivalent formulations of determinacy but for our purposes the most convenient is to use the following:

Definition 2.4.

1. *A tree* σ *is called a* strategy for player I *if all nodes of odd length are totally splitting and all nodes of even length are non-splitting.*
2. *A tree* τ *is called a* strategy for player II *if all nodes of even length are totally splitting and all nodes of odd length are non-splitting.*
3. *A set* $A \subseteq \omega^\omega$ *is called* determined *if there is either a strategy* σ *for player I such that* $[\sigma] \subseteq A$ *or a strategy* τ *for player II such that* $[\tau] \subseteq A^c$.

Since by [So70] it is consistent with ZF that all sets of reals have the regularity properties, the only way to prove a non-trivial pointwise connection between determinacy and these properties is by using AC. The way one would typically prove that there are sets that are, e.g., non-Lebesgue measurable, don't have the Baire property, the perfect set property etc. is by a diagonalization procedure called the *Bernstein construction.* In the most general setting this is the following fact:

Theorem 2.5 (*General Bernstein Theorem*). *Let* $\{X_\alpha \mid \alpha < 2^{\aleph_0}\}$ *be a collection of* $2^{\aleph_0}$ *sets of reals, such that* $|X_\alpha| = 2^{\aleph_0}$ *for all* α. *Then there are disjoint sets* $A, B \subseteq \bigcup_{\alpha<2^{\aleph_0}} X_\alpha$, *called the* Bernstein components, *such that for all* $\alpha < 2^{\aleph_0}$, $X_\alpha \cap A \neq \varnothing$ *and* $X_\alpha \cap B \neq \varnothing$.

3 Marczewski-Burstin Algebras, the Baire Property and Measurability

It is natural to connect each arboreal forcing notion $\mathbb{P}$ to a corresponding regularity property, or a so-called *algebra of measurability*. For example, random forcing (considered as the collection of perfect trees with non-null Lebesgue measure) is naturally connected to Lebesgue-measurability, and Cohen forcing to the Baire property in the standard topology on ω^ω. In analogy with the latter case, Hechler and eventually different forcing are connected to the Baire properties in the dominating and eventually different topologies on ω^ω, respectively.

For the non-topological arboreal forcings, the regularity property usually considered has been the Marczewski-Burstin algebra.

Definition 3.1. *Let* $\mathbb{P}$ *be arboreal and* $A \subseteq \omega^\omega$.

1. *A is called* $\mathbb{P}$-Marczewski-Burstin-measurable *if* $\forall P \in \mathbb{P}\ \exists Q \leq P\ ([Q] \subseteq A\ \vee\ [Q] \subseteq A^{\mathrm{c}})$.
2. *A is called* $\mathbb{P}$-null *if* $\forall P \in \mathbb{P}\ \exists Q \leq P\ [Q] \subseteq A^{\mathrm{c}}$.
3. *A is called* $\mathbb{P}$-meager *if it is a countable union of* $\mathbb{P}$*-null sets.*

We denote the class of $\mathbb{P}$-Marczewski-Burstin-measurable sets by $\mathrm{MB}(\mathbb{P})$, the ideal of $\mathbb{P}$-null sets by $\mathcal{N}_{\mathbb{P}}$ and the σ-ideal of $\mathbb{P}$-meager sets by $\mathcal{I}_{\mathbb{P}}$. Note that when $\mathbb{P}$ is topological then $\mathbb{P}$-null is the same as being *nowhere dense in the* $\mathbb{P}$*-topology* and $\mathbb{P}$-meager is exactly the topological concept of being meager (or *of first category*).

For the standard non-topological forcings $\mathbb{P}$, a fusion argument like in [Je86, p 15 ff] shows that $\mathrm{MB}(\mathbb{P})$ is a σ-algebra. The same holds for Matthias forcing, although the proof is technically more involved (see [El74]). However, for Cohen, Hechler or eventually different forcing, this is not the case: if we let $\mathbb{P}$ be any one of these three forcings, then, for instance, $A := \{x \mid \forall^\infty n\ (x(n) \text{ is even})\}$ is not in $\mathrm{MB}(\mathbb{P})$. To see this, note that for all $P \in \mathbb{P}$ there exists an $x \in [P]$ which is eventually even and a y which is not eventually even, so $[P] \not\subseteq A$ and $[P] \not\subseteq A^{\mathrm{c}}$. On the other hand, we can write $A = \bigcup_N A_N$ where $A_N := \{x \mid \forall n \geq N\ (x(n)$ is even$)\}$, which is easily seen to be $\mathbb{P}$-null. So then $\mathrm{MB}(\mathbb{P})$ is not a σ-algebra, it doesn't contain F_σ sets, and is in general not a regularity property at all.

It is then not at all surprising that in the topological cases, rather than $\mathrm{MB}(\mathbb{P})$ one usually considers the algebra consisting of those sets having the Baire property in the $\mathbb{P}$-topology, which we shall denote by $\mathrm{BP}(\mathbb{P})$. The definition below should shed some light on the precise reason for this dichotomy and the relationship between $\mathrm{MB}(\mathbb{P})$ and $\mathrm{BP}(\mathbb{P})$. It is close to that of the Marczewski-Burstin algebra but is more natural and well-behaved. For example, Ikegami in [Ik08]

uses it to prove general theorems about the strength of projective measurability statements. We shall refer to this property simply by $\mathbb{P}$-measurability.

Definition 3.2. *Let $\mathbb{P}$ be a topological arboreal forcing. For sets A, B we write $A \subseteq^* B$ if $A \setminus B \in \mathcal{I}_{\mathbb{P}}$. Then a set A is called* $\mathbb{P}$-measurable *if*

$$\forall P \in \mathbb{P}\, \exists Q \leq P\ ([Q] \subseteq^* A \text{ or } [Q] \subseteq^* A^{\mathrm{c}}).$$

We shall denote the class of $\mathbb{P}$-measurable sets by $\mathrm{Meas}(\mathbb{P})$.

The following are simple but important properties:

Lemma 3.3. *Let $\mathbb{P}$ be arboreal.*

1. *For all $P \in \mathbb{P}$, $[P]$ is not $\mathbb{P}$-meager,*
2. $\mathrm{MB}(\mathbb{P}) \subseteq \mathrm{Meas}(\mathbb{P})$, *and*
3. $\mathrm{Meas}(\mathbb{P}) = \mathrm{MB}(\mathbb{P})$ *iff $\mathcal{N}_{\mathbb{P}} = \mathcal{I}_{\mathbb{P}}$.*

Proof.

1. Suppose towards contradiction that $[P] = \bigcup_n M_n$ with $M_n \in \mathcal{N}_{\mathbb{P}}$. By induction, let $P_0 \leq P$ s.t. $[P_0] \cap M_0 = \varnothing$. Using the definition of arboreal forcings, let $P_0' \leq P_0$ be anything with a strictly longer stem. Then let $P_1 \leq P_0'$ be s.t. $[P_1] \cap M_1 = \varnothing$, etc. Then we get a sequence
$$P \geq P_0 \geq P_1 \geq P_2 \geq \ldots$$
of trees with strictly increasing stems, hence there is a real $x := \bigcup_n \mathrm{stem}(P_n)$. Moreover, by the general property of trees it is easy to see that $\bigcap_n [P_n] = \{x\}$. So $x \in [P]$ but $x \notin \bigcup_n M_n$: contradiction.
2. Obvious.
3. Suppose $\mathcal{I}_{\mathbb{P}} = \mathcal{N}_{\mathbb{P}}$. If $A \in \mathrm{Meas}(\mathbb{P})$ then for $P \in \mathbb{P}$ there is $Q \leq P$ s.t. $[Q] \cap A \in \mathcal{I}_{\mathbb{P}} = \mathcal{N}_{\mathbb{P}}$ or $[Q] \setminus A \in \mathcal{I}_{\mathbb{P}} = \mathcal{N}_{\mathbb{P}}$. So then there is $R \leq Q$ s.t. $[R] \cap ([Q] \cap A)) = \varnothing$ resp. $[R] \subseteq ([Q] \cap A))$.
   Conversely, let $A \in \mathcal{I}_{\mathbb{P}}$. Since this means that $A \in \mathrm{Meas}(\mathbb{P}) = \mathrm{MB}(\mathbb{P})$, for all P there is $Q \leq P$ such that $[Q] \subseteq A$ or $[Q] \cap A = \varnothing$. But the former is impossible by (1). □

Here, (1) is an analogue of the classical Baire Category Theorem. Results of [Ik08] show that $\mathrm{Meas}(\mathbb{P})$ is always a σ-algebra, and moreover that $\mathrm{Meas}(\mathbb{P}) = \mathrm{BP}(\mathbb{P})$ for topological $\mathbb{P}$. Hence, the difference between the original two properties—$\mathrm{MB}(\mathbb{P})$ and $\mathrm{BP}(\mathbb{P})$—is exactly the difference between "meager" and "nowhere dense". Since from a topological point of view these concepts usually do not coincide, this explains why $\mathrm{MB}(\mathbb{P})$ usually fails to be a good regularity property for topological forcings. (Incidentally, the Ellentuck topology is a well-known example of a topology where "meager" and "nowhere dense" do coincide. Hence, if $\mathbb{P}$ is Matthias forcing then $\mathrm{BP}(\mathbb{P}) = \mathrm{MB}(\mathbb{P})$, and the latter is precisely the collection of *completely Ramsey* sets, *cf.* [El74]).

In [Lö05] the Marczewski-Burstin algebra for non-topological forcing notions (and later it's weak variant) played the crucial role. We shall do the same thing for Meas($\mathbb{P}$). Since the new property is either the same or larger than those previously considered, any statement of the kind "there is a determined set which is not $\mathbb{P}$-measurable" immediately implies the same statement with "$\mathbb{P}$-measurable" replaced by "$\mathbb{P}$-Marcewski-Burstin-measurable" or by "having the Baire property in the $\mathbb{P}$-topology". Thus, our results are a natural generalization of [Lö05].

4 Determinacy and Measurability

Are there determined sets which are not in Meas($\mathbb{P}$)? We get the expected answer: yes. The main ingredient is, as in [Lö05], Bernstein's theorem, but we need a technical argument before we can apply it.

Lemma 4.1. *Let $\mathbb{P}$ be an arboreal forcing notion. If $P \in \mathbb{P}$ and $C \subseteq [P]$ is $\mathbb{P}$-comeager in $[P]$, then there exists a perfect tree T with $[T] \subseteq C$.*

Proof. Let $[P] \setminus C := \bigcup_n M_n$ with each $M_n \in \mathcal{N}_{\mathbb{P}}$. Let $C_n := [P] \setminus M_n$, so that $C = \bigcap_n C_n$. By induction we shall construct a collection of $P_u \in \mathbb{P}$ indexed by $u \in 2^{<\omega}$, while taking care that $[P_u] \subseteq C_{|u|}$ for all u.

- Since $M_0 \in \mathcal{N}_{\mathbb{P}}$, pick $P_\varnothing \leq P$ such that $P_\varnothing \cap M_0 = \varnothing$, i.e., $[P_\varnothing] \subseteq C_0$.
- Suppose we have $u \in 2^{<\omega}$ with $|u| = n$, and $[P_u] \subseteq C_n$. Since P_u is perfect we can extend its stem t to two incompatible stems t' and t''. Since $\mathbb{P}$ is arboreal, there are P'_u and P''_u such that $t' \subseteq \mathrm{stem}(P'_u)$ and $t'' \subseteq \mathrm{stem}(P''_u)$. Now, since $M_{n+1} \in \mathcal{N}_{\mathbb{P}}$, there are $[P_{u^\frown\langle 0\rangle}] \subseteq [P'_u] \setminus M_{n+1}$ and $[P_{u^\frown\langle 1\rangle}] \subseteq [P''_u] \setminus M_{n+1}$.

Let T be the tree generated by $\{\mathrm{stem}(P_u) \mid u \in 2^{<\omega}\}$. By our construction, this is clearly a perfect tree, so it just remains to prove that $[T] \subseteq C$. But, for every $x \in [T]$ there is a $y \in 2^\omega$ such that $x = \bigcup_n \mathrm{stem}(P_{y\restriction n})$. Moreover, it is easy to see that $\bigcap_n [P_{y\restriction n}] = \{x\}$. Therefore, for all n we have $x \in [P_{y\restriction n}] \subseteq C_n$, hence $x \in C$. □

Corollary 4.2. *Let $\mathbb{P}$ be arboreal and $A \subseteq \omega^\omega$ $\mathbb{P}$-measurable. Then*

$$\forall P \in \mathbb{P}\ \exists T \subseteq P\ (T \textit{ is a perfect tree and } [T] \subseteq A \ \textit{ or } \ [T] \subseteq A^{\mathrm{c}}).$$

Proof. Let $A \in \mathrm{Meas}(\mathbb{P})$ and $P \in \mathbb{P}$. We know that there is a $P' \in \mathbb{P}$ with $P' \leq P$ such that $[P'] \setminus A$ is meager or $[P'] \cap A$ is meager. In the former case $C := A \cap [P']$ is comeager so there is a perfect tree in A, and in the latter case $[P'] \setminus A$ is comeager so there is a perfect tree in A^{c}. □

The corollary is sufficient to construct a counterexample using a Bernstein diagonalization procedure:

Theorem 4.3. *Determinacy does not imply $\mathbb{P}$-measurability pointwise.*

Proof. Fix any $P \in \mathbb{P}$ with $|\mathrm{stem}(P)| \geq 2$. Then fix any strategy σ such that $[P] \cap [\sigma] = \varnothing$, which is always possible just by letting the beginning of σ be different from the stem of P. Then let $\langle T_\alpha \mid \alpha < 2^{\aleph_0} \rangle$ be an enumeration of all

perfect trees in $[P]$. By the general Bernstein theorem 2.5 there are disjoint sets $A, B \subseteq \bigcup_{\alpha<2^{\aleph_0}}[T_\alpha] \subseteq [P]$, both of which intersect every T_α. Let $A' := A \cup [\sigma]$. Then, by the contraposition of Corollary 4.2, neither A' nor A'^{c} is in Meas($\mathbb{P}$), but clearly either A' or A'^{c} is determined (the former if σ was a strategy for player I and the latter if it was for player II). □

5 Determinacy and Weak Measurability

In [Lö05], the question became more interesting when instead of full measurability one considered a weak, or local version.

Definition 5.1. *Let $\mathbb{P}$ be arboreal, and let $A \subseteq \omega^\omega$. Then*

1. *A is* weakly $\mathbb{P}$-Marczewski-Burstin-measurable *if $\exists P \in \mathbb{P}$ s.t. $[P] \subseteq A$ or $[P] \subseteq A^{\mathrm{c}}$,*
2. *A is* weakly $\mathbb{P}$-measurable *if $\exists P \in \mathbb{P}$ s.t. $[P] \subseteq^* A$ or $[P] \subseteq^* A^{\mathrm{c}}$.*

We denote the class of weakly $\mathbb{P}$-Marczewski-Burstin-measurable sets by wMB($\mathbb{P}$) and the class of weakly $\mathbb{P}$-measurable sets by wMeas($\mathbb{P}$). An important reason for introducing this property is that it is classwise equivalent to full measurability. By [BrLö99, Lemma 2.1] MB($\mathbb{P}$) and wMB($\mathbb{P}$) are classwise equivalent for all standard $\mathbb{P}$ and all topologically reasonable pointclasses. We will prove the same for Meas($\mathbb{P}$) and wMeas($\mathbb{P}$), plus, we will make precise which condition on $\mathbb{P}$ is required for this equivalence to hold.

Definition 5.2. *Let $\mathbb{P}$ be an arboreal forcing. We say that $\mathbb{P}$ is* topologically homogeneous *if for every $P \in \mathbb{P}$ there is a homeomorphism $f_P : \omega^\omega \xrightarrow{\sim} [P]$, in the sense of the standard topology, such that for every tree T we have $T \in \mathbb{P}$ iff the tree of $f_P[T]$ is in $\mathbb{P}$.*

It can be shown that all the standard examples of arboreal forcing notions $\mathbb{P}$ are topologically homogeneous.

Lemma 5.3. *Let $\mathbb{P}$ be topologically homogeneous and $P \in \mathbb{P}$. Then A is $\mathbb{P}$-meager iff $f_P[A]$ is $\mathbb{P}$-meager.*

Proof. Since f_P is a bijection, it is sufficient to prove the claim for $\mathbb{P}$-meager replaced by $\mathbb{P}$-null. We show that if A is $\mathbb{P}$-null then $f_P[A]$ is $\mathbb{P}$-null—for the converse direction, use f_P^{-1}. Let $Q \in \mathbb{P}$ be arbitrary. We must show that there is an $R \leq Q$ s.t. $[R] \cap f_P[A] = \varnothing$. Since $\mathbb{P}$ is strongly separative, we may assume w.l.o.g. that $Q \leq P$. Then the tree of $f_P^{-1}[Q]$ is a member of $\mathbb{P}$, so by assumption there exists an $R' \leq f_P^{-1}[Q]$ s.t. $[R'] \cap A = \varnothing$. Then let $R :=$ the tree of $f_P[R']$, so $R \leq Q$ and $[R] \cap f_P[A] = \varnothing$. □

Theorem 5.4. *Let $\mathbb{P}$ be a topologically homogeneous arboreal forcing notion and let $\boldsymbol{\Gamma}$ be a pointclass closed under continuous preimages and intersections with closed sets. Then $\boldsymbol{\Gamma} \subseteq$ Meas($\mathbb{P}$) iff $\boldsymbol{\Gamma} \subseteq$ wMeas($\mathbb{P}$).*

Proof. The forward direction is obvious. For the backward direction, let $A \in \boldsymbol{\Gamma}$. Fix a $P \in \mathbb{P}$, and we must show that there is a $Q \leq P$ such that $[Q] \subseteq^* A$

or $[Q] \subseteq^* A^c$. By the assumption on $\boldsymbol{\Gamma}$, we know that $A \cap [P] \in \boldsymbol{\Gamma}$ and hence $A' := f_P^{-1}([A] \cap P) \in \boldsymbol{\Gamma}$. By assumption, there exists a $Q' \in \mathbb{P}$ s.t. $[Q'] \subseteq^* A'$ or $[Q'] \subseteq^* A'^c$. Let Q be the tree of $f_P[Q']$. Then $Q \leq P$ and by Lemma 5.3 $[Q] \subseteq^* A$ or $[Q] \subseteq^* A^c$. □

In [Lö05], the arboreal forcings $\mathbb{P}$ were classified into three groups, in such a way that in the first case determinacy implied wMB($\mathbb{P}$) pointwise, in the second case it did not, and in the third there were examples either way. As we noted in the introduction, we are adopting a stricter definition of arboreal forcing notions which eliminates the pathological examples from [Lö05]. As a result, we are now able to give an exhaustive characterization.

First, we fix an arboreal forcing $\mathbb{P}$. Then we split the situation into two cases:

- **Case 1:** For every strategy σ there exists $P \in \mathbb{P}$ s.t. $P \subseteq \sigma$.
- **Case 2:** For some strategy σ, the set $[\sigma]$ is $\mathbb{P}$-null.

Let us immediately check why this case distinction is exhaustive: suppose Case 1 doesn't hold, so there exists a σ s.t. there is no $P \subseteq \sigma$. But then, for every $P \in \mathbb{P}$ there is a $t \in P \setminus \sigma$ and consequently $Q \leq P$ with $t \subseteq \text{stem}(Q)$. So $[Q] \cap [\sigma] = \varnothing$ and we are in Case 2. Conversely, if $[\sigma]$ is $\mathbb{P}$-null then σ clearly cannot contain any $P \in \mathbb{P}$.

Theorem 5.5. *In* Case 1*, Determinacy implies* wMeas($\mathbb{P}$) *pointwise. In* Case 2*, Determinacy does not imply* wMeas($\mathbb{P}$) *pointwise.*

Proof. *Case 1.* Suppose A is determined. Then there is a strategy σ s.t. $[\sigma] \subseteq A$ or $[\sigma] \subseteq A^c$. It follows immediately that there is a $P \in \mathbb{P}$ s.t. $[P] \subseteq A$ or $[P] \subseteq A^c$, so A is certainly in wMeas($\mathbb{P}$).

Case 2. Fix a strategy σ which is $\mathbb{P}$-null. Let $\mathfrak{T}_{\neg\sigma}$ be the collection of perfect trees disjoint from $[\sigma]$.

Claim. For every $A \in$ wMeas($\mathbb{P}$) there is $T \in \mathfrak{T}_{\neg\sigma}$ such that $[T] \subseteq A$ or $[T] \subseteq A^c$.

Proof. First, suppose there is a $P \in \mathbb{P}$ such that $[P] \subseteq^* A^c$, i.e., $[P] \cap A \in \mathcal{I}_{\mathbb{P}}$. Since $[\sigma]$ is $\mathbb{P}$-null, there is a $Q \leq P$ such that $[Q] \cap [\sigma] = \varnothing$. Then $C := [Q] \setminus A$ is $\mathbb{P}$-comeager in $[Q]$ and disjoint from $[\sigma]$. So by Lemma 4.1 there is a perfect tree $[T] \subseteq C$. Then $[T] \subseteq A^c$ and T is disjoint from $[\sigma]$, so $T \in \mathfrak{T}_{\neg\sigma}$. Now, the case where $[P] \subseteq^* A$ is analogous. □ (Claim)

Since $\mathfrak{T}_{\neg\sigma}$ is a collection of $2^{\aleph_0}$ sets of size $2^{\aleph_0}$, we can use the general Bernstein theorem 2.5 to find disjoint sets A and B intersecting every member of $\mathfrak{T}_{\neg\sigma}$. Note that by construction, both A and B are disjoint from $[\sigma]$. Now let $A' := A \cup [\sigma]$. Then, by the contraposition of the Claim, neither A' nor A'^c is in wMeas($\mathbb{P}$) but clearly either A' or A'^c is determined (again depending on whether σ was a strategy of player I or player II). □

This gives a complete characterization of the pointwise relationship between determinacy and wMeas($\mathbb{P}$). From the standard forcing notions, Sacks and Miller forcing belong to Case 1 while the other forcing notions belong to Case 2.

Note that since $\mathrm{wMB}(\mathbb{P}) \subseteq \mathrm{wMeas}(\mathbb{P})$ for all $\mathbb{P}$, and moreover in Case 1 we have actually proved the stronger result that if A is determined then it is in $\mathrm{wMB}(\mathbb{P})$, we also have a proof of the following:

– In *Case 1*, Determinacy implies $\mathrm{wMB}(\mathbb{P})$ pointwise.
– In *Case 2*, Determinacy does not imply $\mathrm{wMB}(\mathbb{P})$ pointwise.

The reason for the discrepancy with [Lö05] is, as we noted, due to the different definition of arboreal forcings. In [Lö05, p 1243] the author asked "it would be interesting to ask ... whether we can find a natural property of forcings (that all forcings used in applications share) that implies [that all forcing notions $\mathbb{P}$ fall under Case 1 or Case 2]". Thus, our definition of "arboreal forcings" (Definition 2.2) gives a solution to this question.

6 Generalizations to $\mathbb{P}$ vs. $\mathbb{Q}$

Although the original problem, and the conceptual question behind it, was whether determinacy has any pointwise consequences, after proving the above results it became clear that the same methods can be applied, with minimal changes, to the general situation of comparing the measurability algebras of two arboreal forcing notions $\mathbb{P}$ and $\mathbb{Q}$. The generalization of Section 4 is completely straightforward:

Theorem 6.1. *Let $\mathbb{P}$ and $\mathbb{Q}$ be arboreal. Then*

1. $\mathrm{wMB}(\mathbb{P}) \not\subseteq \mathrm{Meas}(\mathbb{Q})$*,*
2. $\mathrm{wMeas}(\mathbb{P}) \not\subseteq \mathrm{Meas}(\mathbb{Q})$*,*
3. $\mathrm{wMB}(\mathbb{P}) \not\subseteq \mathrm{MB}(\mathbb{Q})$*,*
4. $\mathrm{wMeas}(\mathbb{P}) \not\subseteq \mathrm{MB}(\mathbb{Q})$*.*

Proof. Note that by definition of arboreal forcings, it is always possible to find $P \in \mathbb{P}$ and $Q \in \mathbb{Q}$ such that $[P] \cap [Q] = \varnothing$. So fix such P and Q and repeat the construction in Theorem 4.3, with $[\sigma]$ replaced by $[P]$. Then the Bernstein component A (and B) constructed in that proof is not in $\mathrm{Meas}(\mathbb{Q})$ but it is disjoint from $[P]$, hence it is in $\mathrm{wMB}(\mathbb{P})$, which proves 1. Points 2, 3 and 4 follow immediately from 1. □

Note that this includes the case that $\mathbb{P} = \mathbb{Q}$, since we never needed them to be different in the argument. In particular, then, this shows that weak $\mathbb{P}$-measurability is strictly larger than $\mathbb{P}$-measurability, and similarly with the Marczewski-Burstin algebras.

Slightly less trivial is the generalization of Section 5. Here, the following notion is of central importance:

Definition 6.2. *Let $\mathbb{P}$ and $\mathbb{Q}$ be arboreal. We say that* $\mathbb{P}$ is thinner than $\mathbb{Q}$ *if for every $Q \in \mathbb{Q}$ there exists a $P \in \mathbb{P}$ s.t. $P \subseteq Q$.*

In practice, it is always easy to see whether a given $\mathbb{P}$ is thinner than $\mathbb{Q}$: for example, Miller forcing is thinner than Laver forcing but not vice versa, Hechler

forcing is thinner than Cohen forcing but not vice versa, Sacks forcing is thinner than every arboreal forcing etc.

Theorem 6.3. *If* $\mathbb{Q}$ *is thinner than* $\mathbb{P}$ *then* $\mathrm{wMB}(\mathbb{P}) \subseteq \mathrm{wMB}(\mathbb{Q}) \subseteq \mathrm{wMeas}(\mathbb{Q})$. *Otherwise,* $\mathrm{wMB}(\mathbb{P}) \nsubseteq \mathrm{wMeas}(\mathbb{Q})$ *and* $\mathrm{wMB}(\mathbb{P}) \nsubseteq \mathrm{wMB}(\mathbb{Q})$.

Proof. If $\mathbb{Q}$ is thinner than $\mathbb{P}$, the result follows directly. If not, then by the same argument as we have used in Section 5 to prove that Case 1 and Case 2 were exhaustive, it follows that there is a $P \in \mathbb{P}$ such that $[P]$ is $\mathbb{Q}$-null. Then we repeat the construction for Case 2 from Theorem 5.5 with $[\sigma]$ replaced by $[P]$ and get a Bernstein component A such that A is disjoint from $[P]$ and hence in $\mathrm{wMB}(\mathbb{P})$ but $A \notin \mathrm{wMeas}(\mathbb{Q})$, and hence not in $\mathrm{wMB}(\mathbb{Q})$ either. □

Of course, it would be nicer to have a full characterization, in the same vein as above, of $\mathrm{wMeas}(\mathbb{P}) \subseteq \mathrm{wMeas}(\mathbb{Q})$. But this would involve comparing the null-ideals $\mathcal{N}_{\mathbb{P}}$ with $\mathcal{N}_{\mathbb{Q}}$, and the results of [Br95] suggest that there is no general method for doing this.

The only other case that remains, is $\mathrm{Meas}(\mathbb{P}) \subseteq \mathrm{Meas}(\mathbb{Q})$. Again, [Br95] suggests that there is no general method, but we can at least say the following:

Theorem 6.4. *If* $\mathbb{P}$ *is not thinner than* $\mathbb{Q}$*, then* $\mathrm{Meas}(\mathbb{P}) \nsubseteq \mathrm{Meas}(\mathbb{Q})$ *(and even:* $\mathcal{N}_{\mathbb{P}} \nsubseteq \mathrm{Meas}(\mathbb{Q})$*).*

Proof. If $\mathbb{P}$ is not thinner than $\mathbb{Q}$ then, by the argument that we have already seen twice, there exists some $Q \in \mathbb{Q}$ such that $[Q]$ is $\mathbb{P}$-null. Choose this Q, enumerate all perfect trees within $[Q]$ and, as in the proof of Theorem 4.3 find Bernstein components A and B. Then $A \notin \mathrm{Meas}(\mathbb{Q})$ by the contraposition of Corollary 4.2 but $A \subseteq [Q] \in \mathcal{N}_{\mathbb{P}}$, so in particular $A \in \mathrm{Meas}(\mathbb{P})$. □

For example, since Cohen forcing is not thinner than Hechler forcing, there is a set which is nowhere dense in the standard topology but does not have the Baire property in the dominating topology.

Acknowledgements

I would like to thank Jörg Brendle for useful discussions and Benedikt Löwe for many detailed comments, helpful ideas and critical reviews. I would also like to thank Daisuke Ikegami for pointing out Definitions 2.2 and 3.2 and, knowingly or otherwise, improving the content of this paper a great deal.

References

[BaJu95] Bartoszyński, T., Judah, H.: Set Theory, On the Structure of the Real Line. A K Peters (1995)

[Br95] Brendle, J.: Strolling Through Paradise. Fundamenta Mathematicae 148, 2643–2674 (1995)

[BrLö99] Brendle, J., Löwe, B.: Solovay-type characterizations for forcing-algebras. The Journal of Symbolic Logic 64, 1307–1323 (1999)

[El74] Ellentuck, E.: A new proof that analytic sets are Ramsey. Journal of Symbolic Logic 39, 163–165 (1974)

[Ik08] Ikegami, D.: Forcing Absoluteness and Regularity Properties (manuscript, 2008)

[Je86] Jech, T.: Multiple Forcing. Cambridge University Press, Cambridge (1986)

[Ka94] Kanamori, A.: The Higher Infinite, Large Cardinals in Set Theory from Their Beginnings. Springer, Berlin (2004)

[Lö05] Löwe, B.: The pointwise view of determinacy: arboreal forcings, measurability, and weak measurability. Rocky Mountain Journal of Mathematics 35, 1233–1249 (2005)

[So70] Solovay, R.M.: A model of set theory in which every set of reals is Lebesgue measurable. Annals of Mathematics Ser. 2 92, 1–56 (1970)

A Two-Dimensional Hybrid Logic of Subset Spaces

Yi N. Wang

Philosophy Department
Peking University
Beijing, China, 100871
wonease@gmail.com

Abstract. Logics of space typically involve two sorts of entities, points and sets, and so are amenable for investigation using hybrid modal languages with nominals for both sorts. As Hilbert systems for these logics are quite complicated, Gentzen systems are used in this paper, first for the basic two-dimensional hybrid logic and then for the logic of subset spaces, which needs additional rules. This provides a foothold from which to consider extensions to neighborhood and topological logics, and also application fields such as epistemic and doxastic logics.

Keywords: hybrid logic, two-sorted hybrid language, logic of subset spaces, Gentzen system for hybrid logic.

A policewoman measures the speed of a car traveling on a highway with a 120 kph speed limit. The speed of the car is 121 kph and this is also the reading on her velocity radar gun but she does not know whether the car is speeding because the accuracy of the radar is ± 2 kph. She would have known this if the reading had been 122 kph or if she had been using one of the newer models with an accuracy of ± 1 kph.

It is natural to model the policewoman's knowledge about the car's speed as the set of speeds that are compatible with the information she has, which in this case is the interval (119, 123) kph. Her lack of knowledge as to whether the car was speeding is entailed by the existence of speeds within this range that exceed the speed limit and other speeds which do not. If the reading on her radar had been 122 kph, her knowledge would be represented by the interval (120, 124) and she would know that the car was speeding. If she was using the newer model radar, her knowledge would be represented by the interval (120, 122) and again she would know that the car was speeding.

The latter case is interesting because it is an example of how knowledge can be improved by *epistemic effort*. A more searching investigation, with better tools, can increase one's knowledge. The concept of epistemic effort was introduced and discussed by works in the traditions of TOPOLOGIC, originally [1], with a recent survey in [2]. The basic idea is to extend traditional epistemic logic with an operator that quantifies over improvements in the knowledge of the agent, modeling the knowledge as an open set within a topological space, or some

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 196–209, 2009.

generalization thereof. This coincides with the traditional epistemic logic (Cf. [3]), where knowledge is defined in terms of accessibility relations.

[1] and [4] gave an appealing sound and complete two-dimensional logic in which two modal operators K and $\Box$ represent relations between points and sets. $K\varphi$ means that φ holds at every point in the current set and $\Box\varphi$ means that φ holds in every subset of the current set that contains the current point. On the epistemic interpretation, $K\varphi$ means that φ is known and $\Box\varphi$ means that φ holds in every refinement of the current epistemic state. Thus, we can express the policewoman's situation as

$$\Diamond K(\mathit{Speeding}) \wedge \neg K \neg \Diamond K \neg (\mathit{Speeding}).$$

The consideration of the various hypothetical cases is difficult to represent in this limited language. There is no mechanism for moving the set representing the agent's knowledge state to another set that is not a subset, to represent the possibility of a different reading on the radar. And even if we introduced a new modal operator to do this, it would not enable us to reason about specific states. Likewise, to reason about the result of using the newer model radar, we would need a way of referring to a specific refinement rather than merely saying that there is one. Both cases can be covered by allowing names, which seems is the approach adopted here.

Hybrid logic (Cf., say, [5] and [6]) is the result of adding names and other referential mechanisms to modal logic and so is the obvious tool to use to extend the expressive power of topologic. It has a more systematic proof theory than modal logic while retaining the latter's intuitive elegance. Yet some work is required to develop hybrid logic in a two-dimensional setting as well as to capture the relationship between the two dimensions—that of set-membership. In this paper, I will consider two sorts of names, for points and sets respectively, and a two-dimensional satisfaction operator, @.

In the above scenario, the claims about the two hypothetical states (of the old radar reading 122 kph and of the new radar reading 121kpm) can represented as

$$@_{122}^{\pm 2} K(\mathit{Speeding}) \wedge @_{121}^{\pm 1} K(\mathit{Speeding}),$$

the right disjunct of which witnesses $\Diamond K(\mathit{Speeding})$ but is more specific. We can also use

$$@_{110}^{\pm 2} \Box K \neg (\mathit{Speeding})$$

to express something like "driving at the speed of 110 kph cannot lead to being suspected of speeding," which seems a better choice for the driver unless he is pursuing the policewoman.

Other hybrid operators such as a two-dimensional version of the "here-and-now" operator, $\downarrow$, can also be considered. With this, we can represent the same facts as

$$\downarrow_{\mathit{curr\_pnt}}^{\mathit{curr\_set}} . (@_{122}^{\mathit{curr\_set}} K(\mathit{Speeding}) \wedge @_{\mathit{curr\_pnt}}^{\pm 1} K(\mathit{Speeding})).$$

Even more can be done with these extensions to the language. Suppose a policeman brings a radar with the accuracy of ± 1 kph and it happens that he

has also recorded the speed to be 119.5 kph. It may be frustrating that the two of them still cannot make a judgment as to whether the car was speeding, but at least their results are compatible with one another, and this can be represented as

$$@^{\pm 2}_{121} \neg K \neg \Diamond \downarrow^X_x . @^{\pm 1}_{119.5} \neg K \neg \Diamond (x \wedge X).$$

The increased expressive power can also be obtained in other ways. For example, [7] used a *difference modality*, $[\neq]$, to express various topological properties of the space such as "density-in-itself" and "T_1-space," and it is easy to show that these can also be defined using $\downarrow$. In fact, $[\neq]\varphi$ itself can be defined by the hybrid formula

$$[\neq]\varphi := \downarrow s.(s \vee \varphi).$$

This paper will mainly focus on a basic logic of space, called "subset space" logic, in which there is little structure on the sets of points. After an introduction to the languages and semantics, a hybrid system will be presented. Issues arising, including the restriction to topological semantics and the generalization to neighborhood semantics, will be discussed in the final section.

1 The Languages and Semantics

We will consider two sorts of nominals, PNTNOM and SETNOM, which name points and (open) sets respectively.[1] Lower case letters $a, b, c, \ldots$ will be used to stand for members of PNTNOM, while upper case letters $A, B, C, \ldots$ will be used for SETNOM. To introduce the hybrid binder $\downarrow$, there are also corresponding variables: PNTVAR and SETVAR. x and X will be used for members of PNTNOM $\cup$ PNTVAR and SETNOM $\cup$ SETVAR respectively. These various sorts of symbols are grouped together as follows:

NOM = PNTNOM $\cup$ SETNOM   SVAR = PNTVAR $\cup$ SETVAR
PNT = PNTNOM $\cup$ PNTVAR   SET = SETNOM $\cup$ SETVAR
AT = PROP $\cup$ NOM $\cup$ SVAR, PROP for propositional variables.

As a summary of this notation for nominals:

New Atoms	NOM	SVAR
PNT	PNTNOM	PNTVAR
SET	SETNOM	SETVAR

Now we can introduce our languages.

Definition 1 (Two-sorted hybrid languages). *The language* $\mathcal{H}^2(@, \downarrow)$ *is given by the following rule:*

$$\varphi ::= \top \mid p \mid x \mid X \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond\!\!\!\cdot\,\varphi \mid \Diamond\varphi \mid @^X_x \varphi \mid \downarrow^S_s .\varphi$$

[1] To call the sets "open" is suggestive of topology, although subset spaces lack the distinctive features of topological spaces, such as closure under unions and finite intersections.

where $p \in \textsc{prop},\ x \in \textsc{pnt},\ X \in \textsc{set},\ s \in \textsc{pntvar},\ S \in \textsc{setvar}$. *Other connectives or constants, such as* $\vee, \leftrightarrow, \bot$ *are defined as usual. Languages* $\mathcal{H}^2$ *(without* @ *and* $\downarrow$*) and* $\mathcal{H}^2(@)$ *(without* $\downarrow$*) will also be used.*

A *subset space* is a structure (S, Σ) where S is a non-empty domain filled with points and $\Sigma \subseteq \wp S$. A *topological space* is a subset space (T, τ) such that $\emptyset, T \in \tau$ and τ is closed under finite intersection and arbitrary union.

Definition 2 (Topological and subset models). *A structure* (S, Σ, V) *is a* subset model *(*topological model*) if* (S, Σ) *is a subset space (topological space) and* $V : \textsc{prop} \cup \textsc{nom} \to S \cup \wp S$ *is a valuation such that* $V(p) \in \wp S,\ V(a) \in S$ *and* $V(A) \in \Sigma$.

Note that a subset model can be taken as a neighborhood model with the neighborhood function N achieving its value at some point in the domain (the current state is always a witness of the existence). The focus of this paper will be on subset spaces but some further remarks on neighborhood models and topological models will be made in the final section.

The meaning of nominals will be clearer after the following definition of satisfaction.

Definition 3 (Satisfaction). *Given a subset model* $\mathfrak{S} = \langle S, \Sigma, V \rangle$*, assignments* $g_0 : \textsc{pntvar} \to S$ *and* $g_1 : \textsc{setvar} \to \Sigma$*. For any* $t \in S$ *and any* $U \in \Sigma$ *such that* $t \in U$*,*[2]

$\mathfrak{S}, g_0, g_1, t, U \models \top$		*Always*
$\mathfrak{S}, g_0, g_1, t, U \models p$	*iff.*	$t \in V(p)$
$\mathfrak{S}, g_0, g_1, t, U \models a$	*iff.*	$t = V(a)$
$\mathfrak{S}, g_0, g_1, t, U \models A$	*iff.*	$U = V(A)$
$\mathfrak{S}, g_0, g_1, t, U \models s$	*iff.*	$t = g_0(s)$
$\mathfrak{S}, g_0, g_1, t, U \models S$	*iff.*	$U = g_1(S)$
$\mathfrak{S}, g_0, g_1, t, U \models \neg\varphi$	*iff.*	$t \in U \ \ \&\ \ \mathfrak{S}, g_0, g_1, t, U \not\models \varphi$
$\mathfrak{S}, g_0, g_1, t, U \models \varphi \wedge \psi$	*iff.*	$\mathfrak{S}, g_0, g_1, t, U \models \varphi \ \ \&\ \ \mathfrak{S}, g_0, g_1, t, U \models \psi$
$\mathfrak{S}, g_0, g_1, t, U \models \Diamond\!\!\cdot\,\varphi$	*iff.*	$\exists t' \in U.\ \ \mathfrak{S}, g_0, g_1, t', U \models \varphi$
$\mathfrak{S}, g_0, g_1, t, U \models \Diamond\varphi$	*iff.*	$\exists U' \in \Sigma.\ \ (t \in U' \subseteq U \ \ \&\ \ \mathfrak{T}, g_0, g_1, t, U' \models \varphi)$
$\mathfrak{S}, g_0, g_1, t, U \models @_x^X \varphi$	*iff.*	$\mathfrak{S}, g_0, g_1, x^{\mathfrak{S}, g_0}, X^{\mathfrak{S}, g_1} \models \varphi$
$\mathfrak{S}, g_0, g_1, t, U \models \downarrow_s^S.\varphi$	*iff.*	$\mathfrak{S}, g_0[{}^t_s], g_1[{}^U_S], t, U \models \varphi,$

where $p \in \textsc{prop}, a \in \textsc{pntnom}, A \in \textsc{setnom}, s \in \textsc{pntvar}, S \in \textsc{setvar}, x \in \textsc{pnt}, X \in \textsc{set}$.

$\Diamond\!\!\cdot$ and $\Diamond$ are diamond counterparts of $K, \Box$, mentioned above,[3] the former allowing us to go from our current point to a neighbor (in the current set) and

[2] This membership is essential here. We have already assumed $t \in U$ by writing $\mathfrak{S}, g_0, g_1, t, U \models \varphi$. Therefore, only "$\mathfrak{S}, g_0, g_1, t, U \not\models \varphi$" is not enough when interpreting "$\mathfrak{S}, g_0, g_1, t, U \models \neg\varphi$".

[3] I am not going to follow the epistemically motivated notation of [4], preferring differently marked box, $\boxdot$, to "K", if it appears somewhere.

the latter allowing us to shrink our current set to one that contains the current point. $@_x^X$ and $\downarrow_s^S$ are the natural extensions of the standard hybrid operators $@$ and $\downarrow$ to the two-dimensional case.

Note that assignments are only relevant to the satisfaction of sentences (formulas with no free variables) if the sentences contain $\downarrow$. When evaluating the sentences of $\mathcal{H}^2(@)$ they will be omitted.

Now we are ready to talk about the logical systems.

2 Hybrid System $\mathbf{G}_{\mathcal{H}^2(@,\downarrow)}$

Hilbert systems for hybrid logic are quite complicated, especially for the two-dimensional semantics with interaction between the dimensions, as we have in the present case. For this reason, I prefer a simpler, though paper-wasting, sequent calculus. The system $\mathbf{G}_{\mathcal{H}^2(@,\downarrow)}$ is given in Table 1, where the basic part $\mathfrak{Bsc}$ is adapted from [8] with the difference that $@$ and $\downarrow$ are now binary.

We take $\mathbf{G}_{\mathcal{H}^2(@)}$ to be the logical system with language $\mathcal{H}^2(@)$ given by all rules of $\mathbf{G}_{\mathcal{H}^2(@,\downarrow)}$ except $@\downarrow\mathrm{L}$ and $@\downarrow\mathrm{R}$.

Neighborhood rules $\mathfrak{Nbhd}$ characterize familiar semantic properties of points and sets, as their names partly have shown. Pairs of $@_{\text{PROP}}$, $@_{\text{PNT}}$, $@_{\text{SET}}$, $@\diamondsuit'$ and $@\Diamond'$ are all basic facts that hold on subset models.

$@_{\text{PNT}}$ shows that the equivalence of points is not affected by sets, while $@_{\text{PROP}}$ declares that the values of propositional variables only depend on points. $@_{\text{SET}}$ claims that the equivalence of sets is based on points they contain. These three all demonstrate a "bias" towards the dimension of points. In a perspective of logic of cognition, $@\diamondsuit'$ allows an agent's perception of the current point to another point in her perception range, and $@\Diamond'$ reveals an agent's effort to shrink the range, as we have expected.

Rules of Weakening (W) and Contraction (C) are admissible as usual. Also as usual, cut elimination attracts much of our attention.

Theorem 1. *Cut is admissible in* $\boldsymbol{G}_{\mathcal{H}^2(@,\downarrow)}$.

Proof. $\mathfrak{Bsc}$ is not much more than the basic hybrid system. The difference is that nominals are now in two sorts, which does not affect the cut elimination results — cut can be eliminated in a similar way to that used in [8].

I will show the most "involved" case here as an example. This is the case of Seligman-style $@=_{\text{PNT}}$-rules[4] ($@=_{\text{PNT}}\mathrm{L}_0$, $@=_{\text{PNT}}\mathrm{L}_1$, $@=_{\text{PNT}}\mathrm{R}$) in which the cut formula is not the prime formula of the $@=_{\text{PNT}}$-rule but is prime in the other rule:

$$\dfrac{\dfrac{\overset{\mathfrak{P}}{@_x^X y, \Gamma[\tfrac{x}{z}] \longrightarrow \Delta[\tfrac{x}{z}], \varphi[\tfrac{x}{z}]}}{@_x^X y, \Gamma[\tfrac{y}{z}] \longrightarrow \Delta[\tfrac{y}{z}], \varphi[\tfrac{y}{z}]}\,@=_{\text{PNT}}\mathrm{L}_0 \qquad \overset{\mathfrak{Q}}{\varphi[\tfrac{y}{z}], \Gamma' \longrightarrow \Delta'}}{@_x^X y, \Gamma[\tfrac{y}{z}], \Gamma' \longrightarrow \Delta[\tfrac{y}{z}], \Delta'}\,\mathrm{Cut} \qquad \rightsquigarrow$$

[4] Comments towards these rules can be found in [9].

Table 1. System $\mathbf{G}_{\mathcal{H}^2(@,\downarrow)}$

Basic Rules $\mathfrak{Bsc}$

$$(\mathrm{Ax})\ \frac{}{\varphi, \Gamma \longrightarrow \Delta, \varphi} \qquad (@{=}_{\text{PNT}}\mathrm{R})\ \frac{\Gamma \longrightarrow \Delta, @^X_x \top}{\Gamma \longrightarrow \Delta, @^X_x x}$$

$$(@{=}_{\text{PNT}}\mathrm{L}_0)\ \frac{@^X_x y, \Gamma[\genfrac{}{}{0pt}{}{x}{z}] \longrightarrow \Delta[\genfrac{}{}{0pt}{}{x}{z}]}{@^X_x y, \Gamma[\genfrac{}{}{0pt}{}{y}{z}] \longrightarrow \Delta[\genfrac{}{}{0pt}{}{y}{z}]} \qquad (@{=}_{\text{PNT}}\mathrm{L}_1)\ \frac{@^X_x y, \Gamma[\genfrac{}{}{0pt}{}{y}{z}] \longrightarrow \Delta[\genfrac{}{}{0pt}{}{y}{z}]}{@^X_x y, \Gamma[\genfrac{}{}{0pt}{}{x}{z}] \longrightarrow \Delta[\genfrac{}{}{0pt}{}{x}{z}]}$$

$$(@\neg\mathrm{L})\ \frac{\Gamma \longrightarrow \Delta, @^X_x \varphi}{@^X_x \neg\varphi, \Gamma \longrightarrow \Delta} \qquad (@\neg\mathrm{R})\ \frac{@^X_x \varphi, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, @^X_x \neg\varphi}$$

$$(@\wedge\mathrm{L})\ \frac{@^X_x \varphi, @^X_x \psi, \Gamma \longrightarrow \Delta}{@^X_x (\varphi \wedge \psi), \Gamma \longrightarrow \Delta} \qquad (@\wedge\mathrm{R})\ \frac{\Gamma \longrightarrow \Delta, @^X_x \varphi \quad \Gamma \longrightarrow \Delta, @^X_x \psi}{\Gamma \longrightarrow \Delta, @^X_x (\varphi \wedge \psi)}$$

$$(@\dot{\Diamond}\mathrm{L})^{*\star}\ \frac{@^X_x \dot{\Diamond} y, @^X_y \varphi, \Gamma \longrightarrow \Delta}{@^X_x \dot{\Diamond}\varphi, \Gamma \longrightarrow \Delta} \qquad (@\dot{\Diamond}\mathrm{R})^{*}\ \frac{\Gamma \longrightarrow \Delta, @^X_y \varphi \quad \Gamma \longrightarrow \Delta, @^X_x \dot{\Diamond} y}{\Gamma \longrightarrow \Delta, @^X_x \dot{\Diamond}\varphi}$$

$$(@\Diamond\mathrm{L})^{\dagger\ddagger}\ \frac{@^X_x \Diamond Y, @^Y_x \varphi, \Gamma \longrightarrow \Delta}{@^X_x \Diamond\varphi, \Gamma \longrightarrow \Delta} \qquad (@\Diamond\mathrm{R})^{\dagger}\ \frac{\Gamma \longrightarrow \Delta, @^Y_x \varphi \quad \Gamma \longrightarrow \Delta, @^X_x \Diamond Y}{\Gamma \longrightarrow \Delta, @^X_x \Diamond\varphi}$$

$$(@@\mathrm{L})\ \frac{@^X_x \top, @^Y_y \varphi, \Gamma \longrightarrow \Delta}{@^X_x @^Y_y \varphi, \Gamma \longrightarrow \Delta} \qquad (@@\mathrm{R})\ \frac{\Gamma \longrightarrow \Delta, @^Y_y \varphi \quad \Gamma \longrightarrow \Delta, @^X_x \top}{\Gamma \longrightarrow \Delta, @^X_x @^Y_y \varphi}$$

$$(@{\downarrow}\mathrm{L})\ \frac{@^X_x \varphi[\genfrac{}{}{0pt}{}{x\,X}{s\,S}], \Gamma \longrightarrow \Delta}{@^X_x {\downarrow}^S_s.\varphi, \Gamma \longrightarrow \Delta} \qquad (@{\downarrow}\mathrm{R})\ \frac{\Gamma \longrightarrow \Delta, @^X_x \varphi[\genfrac{}{}{0pt}{}{x\,X}{s\,S}]}{\Gamma \longrightarrow \Delta, @^X_x {\downarrow}^S_s.\varphi}$$

Neighborhood Rules $\mathfrak{Nbhd}$

$$(@_{\text{PROP}}\mathrm{L}_0)\ \frac{@^X_x \top, \Gamma \longrightarrow \Delta}{@^X_x p, \Gamma \longrightarrow \Delta} \qquad (@_{\text{PROP}}\mathrm{L}_1)\ \frac{\Gamma \longrightarrow \Delta, @^Y_x \top \quad @^Y_x p, \Gamma \longrightarrow \Delta}{@^X_x p, \Gamma \longrightarrow \Delta}$$

$$(@_{\text{PROP}}\mathrm{R})^{\ddagger}\ \frac{@^Y_x \top, \Gamma \longrightarrow \Delta, @^Y_x p \quad \Gamma \longrightarrow \Delta, @^X_x \top}{\Gamma \longrightarrow \Delta, @^X_x p}$$

$$(@_{\text{PNT}}\mathrm{L}_0)\ \frac{@^X_x \top, \Gamma \longrightarrow \Delta}{@^X_x y, \Gamma \longrightarrow \Delta} \qquad (@_{\text{PNT}}\mathrm{L}_1)\ \frac{\Gamma \longrightarrow \Delta, @^Y_x \top \quad @^Y_x y, \Gamma \longrightarrow \Delta}{@^X_x y, \Gamma \longrightarrow \Delta}$$

$$(@_{\text{PNT}}\mathrm{R})^{\ddagger}\ \frac{@^Y_x \top, \Gamma \longrightarrow \Delta, @^Y_x y \quad \Gamma \longrightarrow \Delta, @^X_x \top}{\Gamma \longrightarrow \Delta, @^X_x y}$$

$$(@_{\text{SET}}\mathrm{L}_0)\ \frac{@^X_x \top, \Gamma \longrightarrow \Delta}{@^X_x Y, \Gamma \longrightarrow \Delta} \qquad (@_{\text{SET}}\mathrm{L}_1)\ \frac{\Gamma \longrightarrow \Delta, @^X_y \top \quad @^Y_y \top, \Gamma \longrightarrow \Delta}{@^X_x Y, \Gamma \longrightarrow \Delta}$$

$$(@_{\text{SET}}\mathrm{L}_2)\ \frac{\Gamma \longrightarrow \Delta, @^Y_y \top \quad @^X_y \top, \Gamma \longrightarrow \Delta}{@^X_x Y, \Gamma \longrightarrow \Delta}$$

$$(@_{\text{SET}}\mathrm{R})^{\star}\ \frac{\Gamma \longrightarrow \Delta, @^X_x \top \quad @^X_y \top, \Gamma \longrightarrow \Delta, @^Y_y \top \quad @^Y_y \top, \Gamma \longrightarrow \Delta, @^X_y \top}{\Gamma \longrightarrow \Delta, @^X_x Y}$$

$$(@\dot{\Diamond}'\mathrm{L})\ \frac{@^X_x \top, @^X_y \top, \Gamma \longrightarrow \Delta}{@^X_x \dot{\Diamond} y, \Gamma \longrightarrow \Delta} \qquad (@\dot{\Diamond}'\mathrm{R})\ \frac{\Gamma \longrightarrow \Delta, @^X_x \top \quad \Gamma \longrightarrow \Delta, @^X_y \top}{\Gamma \longrightarrow \Delta, @^X_x \dot{\Diamond} y}$$

$$(@\Diamond'\mathrm{L})\ \frac{@^Y_x \top, \Gamma \longrightarrow \Delta, @^Y_y \top \quad @^Y_x \top, @^X_y \top, \Gamma \longrightarrow \Delta}{@^X_x \Diamond Y, \Gamma \longrightarrow \Delta}$$

$$(@\Diamond'\mathrm{R})^{\star}\ \frac{\Gamma \longrightarrow \Delta, @^Y_x \top \quad @^Y_y \top, \Gamma \longrightarrow \Delta, @^X_y \top}{\Gamma \longrightarrow \Delta, @^X_x \Diamond Y}$$

* $\varphi \notin$ PNT; * y is new; † $\varphi \notin$ SET; ‡ Y is new.

$p \in$ PROP, $x, y, z \in$ PNT, $X, Y, Z \in$ SET, $s \in$ PNTVAR, $S \in$ SETVAR.

$$\rightsquigarrow \dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{cc}\mathfrak{P}[^x_y] & \mathfrak{Q}[^x_y] \\ @^X_x x, \Gamma[^x_z][^x_y] \longrightarrow \Delta[^x_z][^x_y], \varphi[^x_z][^x_y] & \varphi[^y_z][^x_y], \Gamma'[^x_y] \longrightarrow \Delta'[^x_y]\end{array}}{@^X_x x, \Gamma[^x_z][^x_y], \Gamma'[^x_y] \longrightarrow \Delta[^x_z][^x_y], \Delta'[^x_y]}\,\text{Cut}}{@^X_x y, @^X_x x, \Gamma[^x_z][^x_y], \Gamma'[^x_y] \longrightarrow \Delta[^x_z][^x_y], \Delta'[^x_y]}\,\text{W}\textsc{l}}{@^X_x y, @^X_x y, \Gamma[^y_z], \Gamma' \longrightarrow \Delta[^y_z], \Delta'}\,@{=_{\textsc{pnt}}}\text{L}_0}{@^X_x y, \Gamma[^y_z], \Gamma' \longrightarrow \Delta[^y_z], \Delta'}\,\text{C}\textsc{l}$$

The lengths of $\mathfrak{P}$ and $\mathfrak{Q}$ are not changed after substitution, and Cut can be moved up since $\varphi[^x_z][^x_y]$ is the same as $\varphi[^y_z][^x_y]$.

As for $\mathfrak{Nbhd}$ rules, first note that each pair matches with themselves, and only $@_{\textsc{pnt}}$-rules interact with $@{=_{\textsc{pnt}}}$-rules. Cut still can be moved up in this case. Here I show cut elimination for $(@_{\textsc{pnt}}\text{L}_1, @{=_{\textsc{pnt}}}\text{R})$:

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{P} \\ \Gamma \longrightarrow \Delta, @^X_x \top\end{array}}{\Gamma \longrightarrow \Delta, @^X_x x}\,@{=_{\textsc{pnt}}}\text{R} \qquad \dfrac{\begin{array}{cc}\mathfrak{Q}_0 & \mathfrak{Q}_1 \\ \Gamma' \longrightarrow \Delta', @^Y_x \top & @^Y_x x, \Gamma' \longrightarrow \Delta'\end{array}}{@^X_x x, \Gamma' \longrightarrow \Delta'}\,@_{\textsc{pnt}}\text{L}_1}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\text{Cut}$$

$$\rightsquigarrow \dfrac{\dfrac{\begin{array}{c}\mathfrak{Q}_0 \\ \Gamma' \longrightarrow \Delta', @^Y_x \top\end{array}}{\Gamma' \longrightarrow \Delta', @^Y_x x}\,@{=_{\textsc{pnt}}}\text{R} \qquad \begin{array}{c}\mathfrak{Q}_1 \\ @^Y_x x, \Gamma' \longrightarrow \Delta'\end{array}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\,\text{Cut}$$

Cut is moved up along $\mathfrak{Q}_1$, and $\mathfrak{P}$ is not necessary any more. □

The calculus $\mathbf{G}_{\mathcal{H}^2(@,\downarrow)}$ has the so-called Quasi-Subformula Property. Namely, every formula in premises has only new nominals occurring, or is a genuine subformula of some formula in the conclusion. Some remarks on this issue can be found in [10], although the basis there is natural deduction systems.

Now we are going to show the soundness and completeness of $\mathbf{G}_{\mathcal{H}^2(@,\downarrow)}$ (or more exactly, $\mathbf{G}_{\mathcal{H}^2(@)}$) with respect to all subset models. First is soundness, and completeness will be proved in the next section.

Theorem 2 (Soundness). *All rules of $\mathbf{G}_{\mathcal{H}^2(@)}$ preserve validity in subset models.*

Proof. This is straightforward, given the above discussion of neighborhood rules. As an example I will show that $@\Diamond'\text{R}$ preserves validity.

For any subset model $\mathfrak{S} = (S, \Sigma, V)$, a point $t \in S$ and a set $U \in \Sigma$, assume $\mathfrak{S}, t, U$ satisfies both premises of $@\Diamond'\text{R}$. If $\mathfrak{S}, t, U \not\models \Gamma \longrightarrow \Delta, @^X_x \Diamond Y$, then $\mathfrak{S}, t, U \models \Gamma$ and $\mathfrak{S}, t, U \not\models \Delta, @^X_x \Diamond Y$.[5] Since $\mathfrak{S}, t, U \models @^Y_x \top$ and $\forall y \in \textsc{pnt} : (\mathfrak{S}, t, U \models @^Y_y \top \Rightarrow \mathfrak{S}, t, U \models @^X_y \top)$ hold, according to the two premises, it forces $\mathfrak{S}, V(x), V(Y) \models \top$ and $\forall y \in \textsc{pnt}. (\mathfrak{S}, V(y), V(Y) \models \top \Rightarrow \mathfrak{S}, V(y), V(X) \models \top)$,

[5] For convenience, here I use $\mathfrak{S}, t, U \not\models \Delta, @^X_x \Diamond Y$ to mean that every formula in $\Delta \cup \{@^X_x \Diamond Y\}$ is false at t, U in $\mathfrak{S}$.

which means $V(x) \in V(Y)$ and $V(Y) \subseteq V(X)$. Thus $\mathfrak{S}, t, U \models @_x^X \Diamond Y$ holds, and a contradiction will be reached. $\square$

3 Completeness of $\mathbf{G}_{\mathcal{H}^2(@)}$ for All Subset Models

3.1 Basic Ideas

First consider the @-prefixed sublanguage of $\mathcal{H}^2(@)$, denoted here by $@\mathcal{H}^2(@)$. Formally,

$$@\mathcal{H}^2(@) := \{@_x^X \varphi \mid \varphi \in \mathcal{H}^2(@),\ x \in \text{PNT},\ X \in \text{SET}\}.$$

I will use the Henkin's method to show that $\mathbf{G}_{\mathcal{H}^2(@)}$ is strongly complete with respect to all subset models in the language $@\mathcal{H}^2(@)$. Namely, for every $\varphi \in @\mathcal{H}^2(@)$ and $\Phi \subseteq @\mathcal{H}^2(@)$,

$$\Phi \vdash_{\mathbf{G}_{\mathcal{H}^2(@)}} \varphi \quad \textit{iff.} \quad \Phi \models_{\text{subset}} \varphi.$$

And this result can be easily adapted to the language $\mathcal{H}^2(@)$: For every $\varphi \in \mathcal{H}^2(@)$, $\Phi \subseteq \mathcal{H}^2(@)$,

$$@_x^X \Phi \vdash_{\mathbf{G}_{\mathcal{H}^2(@)}} @_x^X \varphi \quad \textit{iff.} \quad \Phi \vdash_{\mathbf{G}'_{\mathcal{H}^2(@)}} \varphi,$$

where x, X are new nonimals, $@_x^X \Phi$ is a set of @-prefixed formulas, and $\mathbf{G}'_{\mathcal{H}^2(@)}$ is a system with new rules shifting between prefixed and non-prefixed formulas.

3.2 Detailed Proof

We are talking about formulas in the @-prefixed language $@\mathcal{H}^2(@)$ in this subsection if not mentioned explicitly.

Six sorts of formulas of the following forms:

$$@_x^X \diamondsuit\!\!\!\cdot\, \varphi,\ @_x^X \Diamond \psi,\ @_x^X p\ , @_x^X y,\ @_x^X Y,\ @_x^X \Diamond Y \quad (\varphi \notin \text{PNT},\ \psi \notin \text{SET})$$

need witnesses. Let us assume we have already had six enumerations of all those sorts of formulas respectively.

For every consistent set Φ of formulas, we extend it to Φ^+ which contains witnesses for formulas in the above enumerations. Let

$$\begin{aligned}
\alpha_n &= @_{x_n}^{X_n}(@_{x_n}^{X_n} \diamondsuit\!\!\!\cdot\, \varphi_n \to @_{x_n}^{X_n} \diamondsuit\!\!\!\cdot\, z_n \wedge @_{z_n}^{X_n} \varphi_n),\\
\beta_n &= @_{x_n}^{X_n}(@_{x_n}^{X_n} \Diamond \psi_n \to @_{x_n}^{X_n} \Diamond Z_n \wedge @_{x_n}^{Z_n} \psi_n),\\
\gamma_n &= @_{x_n}^{X_n}(@_{x_n}^{X_n} \top \wedge (@_{x_n}^{Z_n} \top \to @_{x_n}^{Z_n} p_n) \to @_{x_n}^{X_n} p_n),\\
\delta_n &= @_{x_n}^{X_n}(@_{x_n}^{X_n} \top \wedge (@_{x_n}^{Z_n} \top \to @_{x_n}^{Z_n} y_n) \to @_{x_n}^{X_n} y_n),\\
\varepsilon_n &= @_{x_n}^{X_n}(@_{x_n}^{X_n} \top \wedge (@_{z_n}^{X_n} \top \to @_{z_n}^{Y_n} \top) \wedge (@_{z_n}^{Y_n} \top \to @_{z_n}^{X_n} \top) \to @_{x_n}^{X_n} Y_n)\\
\zeta_n &= @_{x_n}^{X_n}(@_{x_n}^{Y_n} \top \wedge (@_{z_n}^{Y_n} \top \to @_{z_n}^{X_n} \top) \to @_{x_n}^{X_n} \Diamond Y_n)
\end{aligned}$$

for every $n \in \mathbb{N}$, such that z_n and Z_n are both new in each case.[6] And then define Φ^+ inductively as follows:

$$\begin{aligned} \Phi_0 &= \Phi \\ \Phi_{n+1} &= \Phi_n \cup \{\alpha_n, \beta_n, \gamma_n, \delta_n, \varepsilon_n, \zeta_n\} \\ \Phi^+ &= \textstyle\bigcup_{n\in\mathbb{N}} \Phi_n. \end{aligned}$$

Lemma 1. *Every consistent set of formulas can be extended to a consistent set of formulas that contains witnesses.*

Proof. All we need is to show that the consistency of Φ_n leads to the consistency of Φ_{n+1}. Suppose, for a contradiction, that Φ_{n+1} is inconsistent. Then there exists $\Gamma \subseteq \Phi_n$, such that at least one of

$$\Gamma \cup \{\alpha_n\},\ \Gamma \cup \{\beta_n\},\ \Gamma \cup \{\gamma_n\},\ \Gamma \cup \{\delta_n\},\ \Gamma \cup \{\varepsilon_n\} \text{ or } \Gamma \cup \{\zeta_n\}$$

is inconsistent. (Those witnesses themselves are consistent.) I will take the second case for example to show that is not going to be possible,[7] and omit the subscript n in every occurrence for convenience.

For every φ, given

$$\Gamma, @_y^Y(@_y^Y\Diamond\psi \to @_y^Y\Diamond Z \wedge @_y^Z\psi) \longrightarrow \varphi \tag{1}$$

we can get

$$\Gamma \longrightarrow \varphi, @_y^Y @_y^Y \Diamond\psi \tag{2}$$

and

$$@_y^Y(@_y^Y\Diamond Z \wedge @_y^Z\psi), \Gamma \longrightarrow \varphi. \tag{3}$$

This is because

$$\dfrac{\dfrac{\dfrac{}{@_y^Y @_y^Y\Diamond\psi, (2), @_y^Y(@_y^Y\Diamond Z \wedge @_y^Z\psi)}\text{Ax}}{(2), @_y^Y(@_y^Y\Diamond\psi \to @_y^Y\Diamond Z \wedge @_y^Z\psi)}@{\to}\text{R} \qquad \dfrac{(1)}{@_y^Y(@_y^Y\Diamond\psi \to @_y^Y\Diamond Z \wedge @_y^Z\psi), (2)}\text{WR}}{(2)}\text{Cut}$$

and it is similar for (3). But then

[6] Here we are facing the problem of language expansion. We are using an expanded language each time after adding witnesses, but then new formulas which also need witnesses occur. We expand languages and add new witnesses again and again, and finally the union set of formulas contains witnesses in the union language. (This is fine in an infinite language, as we are using.) I prefer not elaborating this process.

[7] The first case is very similar to the second. All other cases are a bit different because their witnesses are on the left side (as those for universal quantifiers in the first-order predicate logic), but this will not affect us using the same method, because all related rules in $\mathbf{G}_{\mathcal{H}^2(@)}$ have both sides.

$$
\dfrac{(2) \qquad \dfrac{\dfrac{\begin{array}{c}\text{(Easy to prove with } @\wedge\mathrm{R},\ @@\mathrm{R})\\ @^Y_y \Diamond Z, @^Z_y \psi \longrightarrow @^Y_y (@^Y_y \Diamond Z \wedge @^Z_y \psi)\end{array} \qquad (3)}{@^Y_y \Diamond Z, @^Z_y \psi, \Gamma \longrightarrow \varphi}\,\text{Cut}}{@^Y_y @^Y_y \Diamond \psi, \Gamma \longrightarrow \varphi}\,@\Diamond\mathrm{L},\ Z \text{ new}}{\Gamma \longrightarrow \varphi}\,\text{Cut}
$$

A contradiction is reached. □

Then we extend Φ^+ to a maximal consistent set Φ^*. Let $@^{X_0}_{x_0}\varphi_0, @^{X_1}_{x_1}\varphi_1, \dots$ be a list of formulas in Φ^+. Note that we do not add $@^{X_n}_{x_n}\neg\varphi_n$ to Ψ_n in the case that $\Psi_n \cup \{@^{X_n}_{x_n}\varphi_n\}$ is not consistent. This is because there are cases that both $@^{X_n}_{x_n}\varphi_n$ and $@^{X_n}_{x_n}\neg\varphi_n$ do not hold. Actually we have

Proposition 1. *If Φ is consistent, then for every formula $@^X_x\varphi$, one and only one of the three holds:*

$$(1)\ \Phi \vdash @^X_x\varphi, \quad (2)\ \Phi \vdash @^X_x\neg\varphi, \quad (3)\ \Phi \vdash @^X_x\bot.$$

Therefore, Φ^* is achieved through the following process:

$$
\begin{aligned}
\Psi_0 &= \Phi^+,\\
\Psi_{n+1} &= \begin{cases} \Psi_n \cup \{@^{X_n}_{x_n}\varphi_n\}, & \text{if it is } \mathbf{G}_{\mathcal{H}^2(@)}\text{-consistent}\\ \Psi_n \cup \{@^{X_n}_{x_n}\neg\varphi_n\}, & \text{if it is } \mathbf{G}_{\mathcal{H}^2(@)}\text{-consistent}\\ \Psi_n \cup \{@^{X_n}_{x_n}\bot\}, & \text{otherwise}\end{cases}\\
\Phi^* &= \textstyle\bigcup_{n\in\mathbb{N}}\Psi_n.
\end{aligned}
$$

To prove that Φ^* is a maximal consistent set is not much more than a proof of the ordinary Lindenbaum's Lemma (in a countable case here), and Φ^* still contains witnesses since all witnesses are already in Φ^+. Therefore I only state the result as follows while omitting the proof.

Lemma 2. *Every consistent set of formulas can be extended to a maximal consistent set which contains witnesses.* □

Now we define a relation $\sim$ on PNT such that

$$x \sim y \quad :\textit{iff.} \quad \forall X \in \mathrm{SET}.\ (\Phi \vdash @^X_x\top \Rightarrow \Phi \vdash @^X_x y).$$

Lemma 3. *$\sim$ is an equivalence relation.*

Proof. The reflexivity, symmetry and transitivity of $\sim$ are guaranteed by rules $@ =_{\mathrm{PNT}}\mathrm{L}_0$, $@ =_{\mathrm{PNT}}\mathrm{L}_1$, and $@ =_{\mathrm{PNT}}\mathrm{R}$. □

For every point nominal x, let $\overline{x}$ be the equivalence class of it, and for every set nominal X, we define

$$X^\sharp := \{\overline{x} : \Phi \vdash @^X_x\top\}.$$

Let $\mathfrak{S} = (S, \Sigma, V)$ be a structure such that

- S is the set of equivalence classes of all points,
- $\Sigma = \{X^\sharp : X \in \mathrm{SET}\}$,

– V is a valuation such that

$$\begin{aligned} V_c(p) &= \{\overline{x} \in S \mid \forall X \in \text{SET}.\ (\Phi \vdash @_x^X \top \Rightarrow \Phi \vdash @_x^X p)\} \\ V_c(x) &= \overline{x} \\ V_c(X) &= X^\sharp, \end{aligned}$$

where $p \in \text{PROP}, x \in \text{PNT}, X \in \text{SET}$. Clearly $\mathfrak{S}$ is a subset model.

We now prove a Henkin's lemma:

Lemma 4. *For every maximal consistent set Φ of formulas that contains witnesses, every formula φ,*

$$\mathfrak{S}, \overline{z}, Z^\sharp \models \varphi \quad \textit{iff.} \quad \Phi \vdash \varphi,$$

where $\overline{z} \in S, Z^\sharp \in \Sigma$.

Proof. We prove the lemma by induction.

(1) Basic cases ($@_x^X p$, $@_x^X y$, $@_x^X Y$, leaving out the former two cases):

$\varphi = @_x^X Y$: $\mathfrak{S}, \overline{z}, Z^\sharp \models @_x^X Y$ *iff.* $\mathfrak{S}, \overline{x}, X^\sharp \models Y$
iff. $X^\sharp = V_c(Y) = Y^\sharp$ & $\overline{x} \in X^\sharp$
iff. $\forall \overline{y}.\ (\overline{y} \in X^\sharp \Rightarrow \overline{y} \in Y^\sharp)$ & $\forall \overline{z}.\ (\overline{z} \in Y^\sharp \Rightarrow \overline{z} \in X^\sharp)$ & $\overline{x} \in X^\sharp$
iff. $\forall \overline{y}.(\Phi \vdash @_y^X \top \Rightarrow \Phi \vdash @_y^Y \top)$ & $\forall \overline{z}.(\Phi \vdash @_z^Y \top \Rightarrow \Phi \vdash @_z^X \top)$ & $\Phi \vdash @_x^X \top$
iff. $\Phi \vdash @_x^X Y$ ($\Downarrow$: Φ containing witnesses; $\Uparrow$: $@_{\text{SET}}$L,Cut)

(2) Cases for $\neg, \wedge, \diamondsuit, \Diamond, @$ (leaving out cases for $\wedge, \diamondsuit, @$):

$\varphi = @_x^X \neg\psi$: $\mathfrak{S}, \overline{z}, Z^\sharp \models @_x^X \neg\psi$ *iff.* $\mathfrak{S}, \overline{x}, X^\sharp \models \neg\psi$
iff. $\overline{x} \in X^\sharp$ & $\mathfrak{S}, \overline{x}, X^\sharp \not\models \psi$
iff. $\Phi \vdash @_x^X \top$ & $\Phi \nvdash @_x^X \psi$ (Induction Hypothesis (IH))
iff. $\Phi \vdash @_x^X \neg\psi$ (Φ maximal consistent)

$\varphi = @_x^X \Diamond\psi$: $\mathfrak{S}^\Phi, \overline{z}, Z^\sharp \models @_x^X \Diamond\psi$ *iff.* $\mathfrak{S}^\Phi, \overline{x}, X^\sharp \models \Diamond\psi$
iff. $\exists Y^\sharp \in \Sigma_c.\ (\overline{x} \in Y^\sharp \subseteq X^\sharp$ & $\mathfrak{S}^\Phi, \overline{x}, Y^\sharp \models \psi)$
iff. $\exists Y \in \text{SET}.\ (\Phi \vdash @_x^Y \top$ & $(\forall y \in \text{PNT}.\ \Phi \vdash @_y^Y \top \Rightarrow \Phi \vdash @_y^X \top)$ &
& $\Phi \vdash @_x^Y \psi)$ (Defs of $X^\sharp, Y^\sharp$, IH)
iff. $\exists Y \in \text{SET}.\ (\Phi \vdash @_x^X \Diamond Y$ & $\Phi \vdash @_x^Y \psi)$ ($\Downarrow$: Witness; $\Uparrow$: $@\Diamond'$L,Cut)
iff. $\Phi \vdash @_x^X \Diamond\psi$ ($\Downarrow$: $@\Diamond$R; $\Uparrow$: Φ containing witnesses) □

Lemma 4 shows that $\mathfrak{S}$ is exactly the model which we need.

Theorem 3 (Strong completeness). $\boldsymbol{G}_{\mathcal{H}^2(@)}$ *for the language $@\mathcal{H}^2(@)$ is strongly complete with respect to all subset models. That is, for every Φ and φ:*

$$\textit{If } \Phi \models_{subset} \varphi \textit{ then } \Phi \vdash_{\boldsymbol{G}_{\mathcal{H}^2(@)}} \varphi.$$

Proof. The proof follows the routine of Henkin's method after the above lemmas have been shown. □

As is clear, all formulas in $@\mathcal{H}^2(@)$ are prefixed with @s, which is not very elegant. Some tweaks in [8] cover non-prefixed formulas, in which nominals play an important role, and these will be discussed in the next section.

4 Discussions

4.1 Shifting among Prefixed and Non-prefixed Formulas

A simple idea for making the calculus cover non-prefixed formulas is to add labels to every formula, do all inferences with labels in the calculus, and drop the added labels only at the end.

The rule:

$$(\text{Name})\ \frac{@_x^X \varGamma \longrightarrow @_x^X \varDelta}{\varGamma \longrightarrow \varDelta}\ x, X \text{ new}$$

allows every formula being prefixed with a new $@_x^X$ (viewed from bottom up), and then the Gentzen system $\mathbf{G}_{\mathcal{H}^2(@)}$ can be used for deduction.

The Name rule brings in non-prefixed theorem to the system (viewed from top down), but those theorems are merely non-prefixed counterparts, and will not actually express more. This can be made clear by the following theorem.

Theorem 4. *$\mathbf{G}_{\mathcal{H}^2(@)}$+(Name) is sound and strongly complete with respect to all subset models.*

Proof. Soundness can be easily verified by checking the validity-preserving of the new rule. As to the completeness, every semantic consequence $\varPhi \models \varphi$ can be isomorphically changed to

$$@_x^X \varPhi \models @_x^X \varphi,$$

where x, X are new. Then by the completeness in the prefixed language, we get

$$@_x^X \varPhi \vdash @_x^X \varphi.$$

And finally we have $\varPhi \vdash \varphi$, as we want, by the rule Name. □

Adding Name only reaches halfway of the idea given in the beginning of this section. [8] has several new rules to cover direct deduction among non-prefixed formulas, in whose case all non-prefixed rules can be derived. We can also introduce those rules into our system. However, this will not be elaborated in this paper.

4.2 Binary Hybrid Operators vs. Two Sorts of Unary Hybrid Operators

An alternative is to use two unary hybrid satisfaction operators, say $@_x$ and $@_X$, instead of one binary operator. By doing this, many rules are easier to read, especially those that have nothing to do with one kind of nominals, e.g. $@=_{\text{PNT}}$, $@\diamondsuit$ and $@\diamondsuit'$.

Clearly, every binary operator can be reduced to two unary ones between different sorts without affecting the expressive power of the system, and vice versa, under current interpretation.

Using two sorts of unary @s, are we able to make the "syntactical membership" $@_x^X \top$ into the context as it does in the semantics?

Permutation of @s between different sorts currently is not a problem. This is basically because we are using a restricted version of two-dimensions: interpretations of propositional variables only depend on points. This makes $@_x @_X p \equiv @_X @_x p$ true everywhere. But the case without this restriction can also be interesting.

These matters deserve further investigation.

4.3 Generalizations

We are now at a crossroads! In one direction (down), we can think about neighborhood semantics. A structure (W, N, V) is a *neighborhood model* if the following hold:

1. $W \neq \emptyset$
2. $N : W \to \wp\wp W$
3. $V : \textsc{prop} \cup \textsc{nom} \to W \cup \wp W$, where $V(p) \in \wp W,\ V(a) \in W$ and there exists a $w \in W$ such that $V(A) \in N(w)$.

A difference between neighborhood models and subset models is that: arbitrary families of sets are considered in the former, without the requirement that the current point is a member of the current set. But then, we need other facilities to link points and sets. We achieve this by replacing the membership relation $\in$ with another binary relation R, where $RwU \Leftrightarrow U \in N(w)$. Then the semantics for modalities are given as follows:

Let $\mathfrak{M} = (W, N, V)$ be a neighborhood model, $w \in W$ and RwU,

$$
\begin{array}{lll}
\mathfrak{M}, w, U \models \blacklozenge\varphi & \textit{iff.} & Rw\varphi^{\mathfrak{M}} \\
\mathfrak{M}, w, U \models \diamondsuit\varphi & \textit{iff.} & \exists v \in W.\ (RvU \ \ \&\ \ \mathfrak{M}, v, U \models \varphi) \\
\mathfrak{M}, w, U \models \Diamond\varphi & \textit{iff.} & \exists V \subseteq U.\ (RwV \ \ \&\ \ \mathfrak{M}, w, V \models \varphi)
\end{array}
$$

Note that $\blacklozenge$ behaves like the ordinary one-dimensional diamond with respect to neighborhood semantics when RwU holds. $\diamondsuit$ again allows us to jump from one point to another in the same neighborhood, and $\Diamond$ still behaves like a refinement of possible situations. Surely the subset space interpretation can be seen as a special case here, if we take R as $\in$.

To go in the opposite direction (up) from subset spaces to topological spaces seems harder because of the difficulty of expressing the properties of finite intersection and arbitrary union. Adding corresponding mechanism to construct "intersection nominals" or "union nominals" seems a good way to solve the problem. However, that will not be pursued here.

In another direction (sideways?), we can investigate a more abstract account of multi-dimensional hybrid logic. As mentioned in 4.2, restrictions on sets can be removed. SET can be taken as another collection of points, and the membership relation replaced by an arbitrary relation between the two domains. This is characterized precisely by the basic system $\mathfrak{Bsc}$. Abstract multi-dimensional logics with three or even more domains can be considered similarly.

It surely makes sense to go in a fourth direction towards applications in epistemology for which we need a system much weaker than the usual one-dimensional topologic, for which $\Box$ has to be reinterpreted to talk about belief, or for which new modalities have to be added. This is the main motivation of this paper as mentioned in the beginning, although few of these have been actually concerned here.

The hybrid binder $\downarrow$ has not been investigated yet! Much can be done to take it into account.

It is in the above senses that I mean the system in this paper to be taken as foothold, as claimed in the abstract.

References

1. Moss, L.S., Parikh, R.: Topological reasoning and the logic of knowledge. In: Moses, Y. (ed.) Proceedings of the 4th Conference on Theoretical Aspects of Reasoning about Knowledge (TARK), Monterey, CA, pp. 95–105. Morgan Kaufmann, San Francisco (1992) (preliminary report)
2. Parikh, R., Moss, L.S., Steinsvold, C.: Topology and epistemic logic. In: Aiello, M., Pratt-Hartmann, I.E., van Benthem, J.F. (eds.) Handbook of Spatial Logics, pp. 299–341 (2007)
3. Halpern, J.Y., Moses, Y.: Knowledge and common knowledge in a distributed environment. In: PODC 1984: Proceedings of the third annual ACM symposium on Principles of distributed computing, pp. 50–61. ACM, New York (1984)
4. Dabrowski, A., Moss, L.S., Parikh, R.: Topological reasoning and the logic of knowledge. Annals of Pure and Applied Logic 78(1-3), 73–110 (1996)
5. Blackburn, P.: Representation, reasoning, and relational structures: A hybrid logic manifesto. Logic Journal of the IGPL 8(3), 339–365 (2000)
6. Areces, C.E., ten Cate, B.: Hybrid logics. In: Blackburn, P., van Benthem, J.F.A.K., Wolter, F. (eds.) Handbook of Modal Logics. Elsevier, Amsterdam (2006)
7. Kudinov, A.: Topological modal logics with difference modality. In: Hodkinson, I., Venema, Y. (eds.) Advances in Modal Logic, AiML 2006, vol. 6. King's College Publications, London (2006)
8. Seligman, J.: Internalization: The case of hybrid logics. Journal of Logic and Computation 11(5), 671–689 (2001); Special Issue on Hybrid Logics. Areces, C., Blackburn, P. (eds.)
9. Blackburn, P.: Internalizing labelled deduction. Journal of Logic and Computation 10(1), 137–168 (2000)
10. Braüner, T.: Natural deduction for hybrid logics. Journal of Logic and Computation 14(3), 329–353 (2004)

A Unified Framework for Certificate and Compilation for QBF

Igor Stéphan and Benoit Da Mota

LERIA University of Angers, France
{damota,stephan}@info.univ-angers.fr

Abstract. We propose in this article a unified framework for certificate and compilation for QBF. We provide a search-based algorithm to compute a certificate for the validity of a QBF and a search-based algorithm to compile a valid QBF in our unified framework.

1 Introduction

The quantified Boolean formula (QBF) validity problem is a generalisation of the Boolean formulae satisfiability problem. While the complexity of Boolean satisfiability problem is NP-complete, it is PSPACE-complete for the QBF validity problem. This is the price for a more concise representation of many classes of formulae. Many important problems in several research fields have polynomial-time translations to the QBF validity problem : AI planning [1,24], Bounded Model Construction [1], Formal Verification (see [5] for a survey).

Most of the recent decision procedures for QBF validity [11,18,19,28] are extensions of the search-based Davis-Putnam procedure [15] for Boolean satisfiability. A search-based procedure for QBF chooses one Boolean variable, tries to solve two simpler subproblems and combines the results according to the semantics of the quantifier associated to the variable. Some other decision procedures are based on resolution principle (as Q-resolution [9] which extends the resolution principle for Boolean formulae to QBF or Quantor [6] which combines efficiently Q-resolution and expansion), quantifier-elimination algorithms [23,22], or skolemization and SAT solvers [2].

Every finite two-player game can be modeled in QBF [17]. In this kind of applications, a decision procedure (the formula is valid or not) is not sufficient since a solution is needed. A solution of a QBF (a QBF model) is a set of Boolean functions [10]. One possibility to represent them is to build a tree-shape representation (called policy [13] or strategy [7]) but it is exponential in worst case and unfortunately also in usual ones. With a search-based procedure, it is very easy to build a solution of a QBF from the solutions of its two simpler subproblems [7,13].

When a QBF solver returns valid or non-valid, there is no way to check if the answer is correct while in propositional logic the associated result to the decision (a model) is easy to check. A certificate for a valid QBF is any piece of information that provides self-supporting evidence of validity for that QBF [5].

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 210–223, 2009.

A **sat**-certificate [3,4] is a representation of a sequence of Boolean functions for a QBF that certifies its validity. This approach seems to us promising since the generated certificate is not linked to the representation of the input QBF but only to its semantics. The computation of a **sat**-certificate is described in [2] in the framework of sKizzo as a reconstruction from a trace. To the best of our knowledge, there is no result of how to build a **sat**-certificate of a QBF from the **sat**-certificates of its two simpler subproblems. It is an important issue since most of the QBF solvers are search-based procedures. Our first contribution is double (Section 4): we define an operator for **sat**-certificates in order to be able to build a **sat**-certificate of a QBF from the **sat**-certificates of its two simpler subproblems and we describe an algorithm which extends any search-based algorithm to build a **sat**-certificate for a valid QBF during the decision process and not a posteriori from a trace.

In general, a knowledge base is compiled off-line into a target language which is then used on-line to answer some queries. In QBF case, seen as a two-player game, one of the most useful query for the existential player is : what should I play to still be sure to win? Our second contribution is a unified framework for certificate and compilation of QBF (Section 3): the literal base representation which is an extension of the **sat**-certificate representation. In order to extend any search-based procedure to a QBF compiler, an important issue is how to compute the compilation of a QBF from the result of the compilation of its two simpler subproblems. Our third contribution is also double (Section 5): we define an operator for literal bases in order to be able to compute the compilation of a QBF from the literal bases of its two simpler subproblems and we describe an algorithm which extends any search-based algorithm to compile a valid QBF.

Finally we discusse related work (Section 6) and we draw a conclusion (Section 7).

2 Preliminaries

2.1 Propositional Logic

The set of propositional variables is denoted by $\mathcal{V}$. Symbols $\bot$ and $\top$ are the propositional constants ($\overline{\top} = \bot$ and $\overline{\bot} = \top$). Symbol $\wedge$ stands for conjunction, $\vee$ for disjunction, $\neg$ for negation, $\rightarrow$ for implication and $\leftrightarrow$ for bi-implication. A literal is a propositional variable or the negation of a propositional variable. A formula is in conjunctive normal form (CNF) if it is a conjunction of disjunctions of literals. Definitions of the language of propositional formula **PROP** and semantics of all the Boolean symbols are defined in standard way. A substitution is a function from propositional variables to **PROP**. This definition is extended as usual to a function from **PROP** to **PROP**: $[x \leftarrow F](G)$ is the formula obtained from G by replacing occurrences of the propositional variable x by the formula F. This definition is also extended as usual for the substitution of a formula by another formula. An interpretation v is a function from $\mathcal{V}$ to $\{\mathbf{true}, \mathbf{false}\}$; the extension to **PROP** is denoted v^*. To an interpretation v is associated a set in the standard way. Propositional satisfaction is denoted $\models$

($v \models F$ means $v^*(F) = \textbf{true}$, the propositional formula F is satisfied by v and v is a model of F). Logical equivalence is denoted $\equiv$. To a Boolean function f (i.e. a function from $\{\textbf{true}, \textbf{false}\}^n$ to $\{\textbf{true}, \textbf{false}\}$) is associated a propositional formula ψ_f on the variables $\{x_1, \ldots, x_n\}$ such that $v^*(\psi_f) = \textbf{true}$ if and only if $f(v(x_1), \ldots, v(x_n)) = \textbf{true}$ for any interpretation v.

2.2 Quantified Boolean Formulae

The symbol $\exists$ stands for the existential quantifier and $\forall$ stands for the universal quantifier(q stands for any quantifier). A binder Q is a string $q_1 x_1 \ldots q_n x_n$ with $x_1, \ldots, x_n$ distinct propositional variables and $q_1 \ldots q_n$ quantifiers. It is assumed that distinct quantifiers bind occurrences of distinct propositional variables. The empty string is denoted by ϵ. A (prenex) quantified Boolean formula (QBF) is constituted of a binder and a propositional formula called the matrix (only closed formulae are considered: each variable in the matrix is also in the binder). A QBF is in conjunctive normal form if its matrix is in conjunctive normal form. The semantics of QBF is defined as follows: for every propositional variable y and every QBF QM

$$\exists y QM = (Q[y \leftarrow \top](M) \vee Q[y \leftarrow \bot](M))$$

and

$$\forall y QM = (Q[y \leftarrow \top](M) \wedge Q[y \leftarrow \bot](M)).$$

A QBF F is valid if $F \equiv \top$. If y is an existentially quantified variable preceded by the universally quantified variables $x_1, \ldots, x_n$ we denote $\hat{y}_{x_1 \ldots x_n}$ its Skolem function from $\{\textbf{true}, \textbf{false}\}^n$ to $\{\textbf{true}, \textbf{false}\}$. A model for a valid QBF QM is a sequence $\psi_{\hat{y}_1}; \ldots; \psi_{\hat{y}_p}$ such that $[y_1 \leftarrow \psi_{\hat{y}_1}] \ldots [y_p \leftarrow \psi_{\hat{y}_p}](M)$ is a tautology [10] ($y_1, \ldots, y_p$ the existentially quantified variables of QM). For example, the QBF $\exists a \exists b \forall c((a \vee b) \leftrightarrow c)$ is not valid (since $\exists a \exists b \forall c((a \vee b) \leftrightarrow c) \equiv \exists a \exists b(((a \vee b) \leftrightarrow \top) \wedge ((a \vee b) \leftrightarrow \bot)) \equiv \bot$) but the QBF $\forall c \exists a \exists b((a \vee b) \leftrightarrow c)$ is valid and one of its possible model is $\psi_{\hat{a}}; \psi_{\hat{b}}$ with $\psi_{\hat{a}} = c$ and $\psi_{\hat{b}} = \bot$ (since $[a \leftarrow \psi_{\hat{a}}][b \leftarrow \psi_{\hat{b}}]((b \vee a) \leftrightarrow c) = ((\bot \vee c) \leftrightarrow c)$ is a tautology). A (Boolean) model of an unquantified Boolean formula corresponds exactly to a (QBF) model of its existential closure. A QBF is valid if and only if there exists a model. An interpretation v is said to be in a model $\psi_{\hat{y_1}}; \ldots; \psi_{\hat{y_p}}$ if for every i, $1 \leq i \leq n$, $v^*(\psi_{\hat{y_i}}) = v(y_i)$; for example the interpretation $v = \{c, a, \neg b\}$ is in the above model since $v^*(\psi_{\hat{a}}) = \textbf{true} = v(a)$ and $v^*(\psi_{\hat{b}}) = \textbf{false} = v(b)$ but $v' = \{\neg c, a, \neg b\}$ is not in it since $v'^*(\psi_{\hat{a}}) = \textbf{false} \neq v'(a)$. Model-equivalence for QBF is defined in [25] as follows : Two QBF F and F' are model-equivalent (denoted $F \cong F'$) if every model of F is a model of F' and conversely; this equivalence is about preservation of models and not only preservation of validity; for example, $\forall c \exists a \exists b((a \vee b) \leftrightarrow c) \equiv \top$ but $\forall c \exists a \exists b((a \vee b) \leftrightarrow c) \ncong \top$.

2.3 sat-certificate

A **sat**-certificate [3] for a QBF F, with $y_1, \ldots, y_p$ its existentially quantified variables, is a sequence of pairs of formulae $(\phi_1, \nu_1); \ldots; (\phi_p, \nu_p)$, ϕ_i and ν_i defined over the universally quantified variables of F preceding the variable y_i,

$1 \leq i \leq p$. It is defined in [3] only for CNF QBF with sequences of pairs of binary decision diagrams (BDD) [8]. A **sat**-certificate is consistent if for every i, $1 \leq i \leq p$, $(\phi_i \wedge \nu_i) \equiv \bot$. To certify the validity of a CNF QBF QM with a **sat**-certificate $(\phi_1, \nu_1); \ldots; (\phi_p, \nu_p)$ we have to check if $[\neg x_1 \leftarrow \nu_1][x_1 \leftarrow \phi_1] \ldots [\neg x_p \leftarrow \nu_p][x_p \leftarrow \phi_p](M)$ is a tautology. If the verification fails either the QBF is non valid or the **sat**-certificate is not correct; conversely, if the verification succeeds then the QBF is valid and the **sat**-certificate encodes a model [3]. For example, from [3], the sequence of pairs of formulae $(\phi_c, \nu_c); (\phi_e, \nu_e); (\phi_f, \nu_f)$ with $\phi_c = \neg a$, $\nu_c = a$, $\phi_e = (a \wedge b \wedge d) \vee (\neg a \wedge \neg d)$, $\nu_e = (\neg a \wedge d) \vee (a \wedge \neg d)$, $\phi_f = (a \wedge b \wedge \neg d) \vee (\neg a \wedge b \wedge d)$ and $\nu_f = (\neg a \wedge \neg d)$ is a **sat**-certificate for the CNF QBF

$$\xi = \forall a \forall b \exists c \forall d \exists e \exists f \mu \tag{1}$$

with

$$\begin{aligned} \mu = [&(\neg b \vee e \vee f) \wedge (a \vee c \vee f) \wedge (a \vee d \vee e) \wedge (\neg a \vee \neg b \vee \neg d \vee e) \wedge (\neg a \vee b \vee \neg c) \\ &\wedge (\neg a \vee \neg c \vee \neg f) \wedge (a \vee \neg d \vee \neg e) \wedge (\neg a \vee d \vee \neg e) \wedge (a \vee \neg e \vee \neg f)]. \end{aligned}$$

This **sat**-certificate certifies the validity of this CNF QBF since $[\neg c \leftarrow \nu_c][c \leftarrow \phi_c][\neg e \leftarrow \nu_e]$ $[e \leftarrow \phi_e][\neg f \leftarrow \nu_f][f \leftarrow \phi_f](M)$ is a tautology. One can remark that this **sat**-certificate is consistent and that

$$\phi_c; \phi_e; \phi_f = \neg a; (a \wedge b \wedge d) \vee (\neg a \wedge \neg d); (a \wedge b \wedge \neg d) \vee (\neg a \wedge b \wedge d) \tag{2}$$

and $\neg \nu_c; \neg \nu_e; \neg \nu_f = \neg a; \neg(\neg a \wedge d) \vee (a \wedge \neg d); \neg(\neg a \wedge \neg d)$ are a two different models for the QBF ξ.

3 Literal Base

In this section we present formally our proposal for a unified framework for certificate and compilation for QBF: the literal base representation. This representation extends the **sat**-certificate representation of [3].

Definition 1 (Literal base). *A* literal base *is a pair (Q, G) constituted*

- *either of $Q = \epsilon$ and $G = \top$ or $G = \bot$;*
- *either of a binder $Q = q_1 x_1 \ldots q_n x_n$, $n > 0$, and a sequence of pairs of formulae $G = (P_1, N_1); \ldots; (P_n, N_n)$ such that the formulae P_k and N_k, called* guards*, are only built on the variables $\{x_1, \ldots, x_{k-1}\}$ (or $\top$ or $\bot$ when $k = 1$).*

We denote $\mathcal{B}_Q$ the set of the literal bases for a binder Q, $LB = \bigcup_Q \mathcal{B}_Q$ the literal base language and define the function grds such that $grds((Q, G)) = G$.

A literal base is an explicit representation in the order of the binder of the dependencies that have to exist between an existentially quantified variable and the variables preceding it.

By the latter definition:

- if $Q = \epsilon$ then $\mathcal{B}_\epsilon = \{(\epsilon, \top), (\epsilon, \bot)\}$;
- if $Q = qx$ then $\mathcal{B}_{qx} = \{(qx, (\top, \top)), (qx, (\top, \bot)), \ \ (qx, (\bot, \top)), (qx, (\bot, \bot))\}$

If n is the number of variables of a binder Q then the size of $\mathcal{B}_Q$ is $\underbrace{2^{2^{\cdots^{2}}}}_{n+1}$.

We interpret a literal base language as a representation for a subset of the QBF language.

Definition 2 (Interpretation of a literal base). *The* interpretation function *is a function from LB to QBF denoted* $(.)^*$ *and is defined as follows :*

- *if* $lb = (\epsilon, G)$ *then* $lb^* = G$ *;*
- *if* $lb = (q_1x_1 \dots q_nx_n, (P_1, N_1); \dots; (P_n, N_n)), n > 0$*, then*

$$lb^* = q_1x_1 \dots q_nx_n \bigwedge_{k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k))$$

If X is a subset of BL then X^* denotes $\{lb^* | lb \in X\}$. From here *non_valid* denotes a literal base whose interpretation is non valid and we extend the latter definition by $non\_valid^* = \bot$.

Clearly enough from Definition 2, if a literal base $(Q, (P_1, N_1); \dots; (P_n, N_n))$ is such that its interpretation is valid then necessarily for every universally quantified variable x_i, P_i and N_i can be replaced in the literal base by $\top$. The following literal base

$$\beta = (\forall a \forall b \exists c \forall d \exists e \exists f, (\top, \top); (\top, \top); (P_c, N_c); (\top, \top); (P_e, N_e); (P_f, N_f)) \quad (3)$$

with $P_c = \neg a$, $N_c = a$, $P_e = (\neg d \wedge c \wedge \neg a) \vee (d \wedge \neg c \wedge a)$,

$$N_e = (d \wedge c \wedge \neg a) \vee (\neg c \wedge \neg b \wedge a) \vee (\neg d \wedge \neg c \wedge a),$$

$$P_f = (\neg e \wedge d \wedge c \wedge \neg a) \vee (\neg e \wedge \neg c \wedge \neg b \wedge a) \vee (d \wedge \neg c \wedge \neg b \wedge a) \vee (\neg e \wedge \neg d \wedge \neg c \wedge a) \vee (e \wedge d \wedge \neg c \wedge a)$$

and

$$N_f = (\neg e \wedge d \wedge c \wedge \neg b \wedge \neg a) \vee (e \wedge \neg d \wedge c \wedge \neg a) \vee (\neg e \wedge \neg c \wedge \neg b \wedge a) \vee (d \wedge \neg c \wedge \neg b \wedge a) \vee (e \wedge d \wedge \neg c \wedge a),$$

is such that its interpretation is model-equivalent to (1) (i.e. $\beta^* \cong \xi$).

The following theorem establishes that for every QBF there exists a literal base such that its interpretation is model-equivalent to the QBF. By this theorem the literal base language may be considered as a target compilation language for QBF.

Theorem 1 (Completeness of LB**).** *Let QM be a QBF. Then there exists a literal base* $lb \in \mathcal{B}_Q$ *such that* $lb^* \cong QM$*.*

A **sat**-certificate for a QBF is easily extended to a literal base: we add the binder of the QBF and in the sequence of the **sat**-certificate for each universally quantified variable we add a couple $(\top, \top)$. Hence, the interpretation of the **sat**-certificate considered as a literal base has only one model which is the model of the QBF. In a **sat**-certificate $(\phi_1, \nu_1); \ldots; (\phi_p, \nu_p)$ formulae ϕ_i and ν_i, $1 \leq i \leq p$, only depend on the preceding universally quantified variables while in a literal base $(Q, (P_1, N_1); \ldots; (P_n, N_n))$ formulae P_i and N_i, $1 \leq i \leq n$, may depend on all the preceding variables.

The propositional fragment in which the propositional formulae of the literal bases are defined needs only to be complete and may be chosen w.r.t. its succinctness (see [14] for a survey on properties of propositional fragments).

When a QBF is considered to model a finite two-player game, the validity of the QBF means that the "existential" player is sure to win if he follows the moves obtained from the (sequence of formulae of the) model. We are interested in the following question: since until now I have followed a (sequence of formulae of a) model, can I change my mind for the next move? We call this problem the "next move choice problem" and we define it formally.

Definition 3 (Next move choice problem for a subset X of QBF).

- Instance : *A formula $q_1x_1 \ldots q_nx_nM$ from a subset X of QBF, a sequence of substitutions $[x_1 \leftarrow C_1] \ldots [x_i \leftarrow C_i]$ obtained from a (sequence of formulae of a) model for $q_1x_1 \ldots q_nx_nM$ with $q_i = \exists$ and $C_1, \ldots, C_i \in \{\top, \bot\}$.*
- Query: *Does there exist a model for* $q_{i+1} \ldots q_nx_n[x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \overline{C_i}](M)$.

Clearly enough, the next move choice problem is still PSPACE-complete if we consider $X = QBF$.

Considering again the QBF (1) and one of its model (2), we know that $\forall d \exists e \exists f[a \leftarrow \top][b \leftarrow \top][c \leftarrow \bot](\mu)$ is valid (since $[a \leftarrow \top][b \leftarrow \top](\phi_c) \equiv \bot$) but is $\forall d \exists e \exists f[a \leftarrow \top][b \leftarrow \top][c \leftarrow \top](\mu)$ also valid?

We introduce a new property, called "optimality", for literal bases in order to exhibit a QBF fragment in which the next move choice problem is polytime w.r.t the size of the literal base.

Definition 4 (Optimality of a literal base). *Let lb be a literal base such that $lb = (q_1x_1 \ldots q_nx_n, (P_1, N_1); \ldots; (P_n, N_n))$ and $lb^* = q_1x_1 \ldots q_nx_nM$. The literal base lb is optimal if the following holds. For all i, $1 \leq i \leq n$, let $[x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}]$ be an interpretation such that for all k, $1 \leq k < i$ if $C_k = \top$ then $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](P_k)$ else $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](N_k)$.*

Then

$$\models [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}](P_i)$$
if and only if there exists a model for
$$q_{i+1}x_{i+1} \ldots q_nx_n[x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \top](M)$$

and

$$\models [x_1 \leftarrow C_1] \dots [x_{i-1} \leftarrow C_{i-1}](N_i)$$
if and only if there exists a model for
$$q_{i+1}x_{i+1} \dots q_n x_n [x_1 \leftarrow C_1] \dots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \bot](M).$$

We denote by OBL the set of optimal literal bases.

Considering again (3), β is an optimal literal base. Since the interpretation of β is model-equivalent to (1) (i.e. $\forall a \forall b \exists c \forall d \exists e \exists f \mu$) and $[a \leftarrow \top][b \leftarrow \top](N_c) \equiv \bot$ the QBF $\forall d \exists e \exists f [a \leftarrow \top][b \leftarrow \top][c \leftarrow \top](\mu)$ is not valid.

The most important property of optimal literal bases is that the next move choice problem is polytime and no more PSPACE-complete.

Theorem 2. *The next move choice problem for OBL^* is polytime w.r.t. the size of the literal base.*

If a QBF modeling a finite two-player game is compiled off-line in an optimal literal base, the computation of any sequence of moves leading to victory is polytime. An optimal literal base may be seen as a dynamic decision tree. The property of optimality of a literal base is linked with the property of minimality of a QBF which expresses that the QBF matrix contains only the models needed by the (QBF) models.

Definition 5 (Minimality of a QBF). *A QBF is minimal if all the (propositional) models of the matrix are (at least) in one of its (QBF) model.*

For example, the QBF $\exists a \forall b ((a \wedge b) \vee (a \wedge \neg b) \vee (\neg a \wedge b))$ is not minimal since the (Boolean) model $\{\neg a, b\}$ of the matrix is not in the only one model $\psi_{\hat{a}} = \top$.

Theorem 3. *Let lb be an optimal literal base. Then lb^* is a minimal QBF.*

The converse of Theorem 3 is false: The literal base $(\exists a \forall b, (\top, \top), (a, a))$ is not optimal (since there is no model with $\psi_{\hat{a}} = \bot$) but its interpretation is minimal.

4 Literal Base and sat-certificate for Search-Based Algorithms

In this section we are interested in the following problem: how to extend a search-based procedure in order to compute directly the **sat**-certificate and not a posteriori from a trace. To do this we define an operator for literal bases in order to be able to build a **sat**-certificate from the **sat**-certificates of its two simpler subproblems.

Definition 6. *The operator $\circ_x : \mathcal{B}_Q \times \mathcal{B}_Q \rightarrow \mathcal{B}_{\forall x Q}$ is defined as follows :*

$$\begin{aligned}
&(Q, (P_1, N_1); \dots; (P_n, N_n)) \circ_x (Q, (P'_1, N'_1); \dots; (P'_n, N'_n)) \\
&= (\ \forall x Q, (\top, \top); \\
&\quad (((\neg x \vee P_1) \wedge (x \vee P'_1)), ((\neg x \vee N_1) \wedge (x \vee N'_1))); \dots; \\
&\quad (((\neg x \vee P_n) \wedge (x \vee P'_n)), ((\neg x \vee N_n) \wedge (x \vee N'_n))))
\end{aligned}$$

In this definition, if x is interpreted to **true** (resp. **false**) then for all i, $1 \leq i \leq n$, $((\neg x \vee P_i) \wedge (x \vee P'_i)) \equiv P_i$ (resp. P'_i) and $((\neg x \vee N_i) \wedge (x \vee N'_i)) \equiv N_i$ (resp. N'_i). If $(Q, (P_1, N_1); \ldots; (P_n, N_n))$ and $(Q, (P'_1, N'_1); \ldots; (P'_n, N'_n))$ are **sat**-certificates and $Q = q_1 x_1 \ldots q_n x_n$ with $q_i = \forall$ then clearly enough $((\neg x \vee P_i) \wedge (x \vee P'_i)) \equiv \top \equiv ((\neg x \vee N_i) \wedge (x \vee N'_i))$.

We establish by the following theorem that the $\circ$ operator composes two **sat**-certificates in a new **sat**-certificate.

Theorem 4. *Let $\forall x Q M$ be a QBF. If $lb_\top$ is a **sat**-certificate for $Q[x \leftarrow \top](M)$ and $lb_\perp$ is a **sat**-certificate for $Q[x \leftarrow \perp](M)$ then $(lb_\top \circ_x lb_\perp)$ is a **sat**-certificate for $\forall x Q M$.*

Algorithm 1. *search_certif_qbf*

In: Q : a binder of a QBF
In: M : a matrix of a QBF
Out: a **sat**-certificate or *non_valid*

```
if Q = qx then
  if q = ∃ then
    switch M do
      case ⊤ : return (∃x, (⊤, ⊥))
      case ⊥ : return non_valid
      case x : return (∃x, (⊤, ⊥))
      case ¬x : return (∃x, (⊥, ⊤))
    end switch
  else
    if M ≡ ⊤ then return (∀x, (⊤, ⊤)) else return non_valid end if
  end if
else
  Q = qxQ'
  lb+ := search_certif_qbf(Q', M[x ← ⊤])
  if lb+ = non_valid then
    if q = ∃ then
      lb- := search_certif_qbf(Q', M[x ← ⊥])
      if lb- = non_valid then return non_valid
      else return ((Q, (⊥, ⊤) ; grds(lb-)) end if
    else
      return non_valid
    end if
  else
    if q = ∃ then
      return (Q, (⊤, ⊥) ; grds(lb+))
    else
      lb- := search_certif_qbf(Q', M[x ← ⊥])
      if lb- = non_valid then return non_valid else return (lb+ ∘x lb-) end if
    end if
  end if
end if
```

We are now able to present the search-based algorithm $search\_certif\_qbf$ which computes a **sat**-certificate for a QBF. The $search\_certif\_qbf$ algorithm checks first if the binder is reduced to a single quantifier with its variable. In this case, if it is an existential quantifier four cases are possible, corresponding, in the order of the algorithm, to : $\exists x\top \equiv \exists x x^1$, $\exists x\bot \equiv \bot$, $\exists x x \cong \exists x((\neg x\vee\top)\wedge(x\vee\bot))$ and $\exists x\neg x \cong \exists x((\neg x\vee\bot)\wedge(x\vee\top))$. If the quantifier is universal then if $M \equiv \top$ then $\forall x M \equiv \top$ else $\forall x x \equiv \forall x\neg x \equiv \forall x\bot \equiv \bot$. If there are some quantifiers, since the algorithm is a search-based one, the most external quantifier is considered. If this quantifier is existential then if one of the recursive calls for the substitution by $\top$ (resp. $\bot$) for the variable x is different to non_valid the returned **sat**-certificate is $(Q, (\top, \bot); grds(lb^+))$ (resp. $(Q, (\bot, \top); grds(lb^-))$) which expresses that x must be **true** (resp. **false**). If the quantifier is universal then if at least one recursive call for the substitution by $\top$ or by $\bot$ for the variable x returns non_valid then non_valid is returned otherwise the Skolem functions of the two **sat**-certificates have to be combined to integrate the new argument x by $(lb^+ \circ_x lb^-)$ before this new **sat**-certificate is returned.

Theorem 5 (Correctness of $search\_certif\_qbf$). *Let QM be a QBF. $search\_certif\_qbf(Q, M)$ returns a **sat**-certificate for QM if the QBF is valid and non_valid otherwise.*

In case of search-based algorithms for CNF QBF, unit propagation and monotone literal propagation [11] may be easily added to the $search\_certif\_qbf$ algorithm.

5 Literal Bases and QBF Compilation for Search-Based Algorithms

Since Theorem 1 establishes the completeness of the literal base language, LB may be considered as a target language for the compilation of a QBF. In this section we are interested in the following problem: how to extend a search-based procedure in order to compile a QBF in an optimal literal base. To do this we define an operator for literal bases which compile a QBF by the combination of the results of the compilation of its two simpler subproblems.

Definition 7. *Let $Q' = q_2x_2 \ldots q_nx_n$ and $Q = q_1x_1Q'$ be two binders and $lb, lb' \in \mathcal{B}_Q$. The operator $\oplus : \mathcal{B}_Q \times \mathcal{B}_Q \to \mathcal{B}_Q$ is defined as follows :*
If $Q = \epsilon$ then $(lb \oplus lb') = (lb^*\vee lb'^*)$ else

$$\begin{aligned}
&(Q, (P_1, N_1); \ldots; (P_n, N_n)) \oplus (Q, (P'_1, N'_1); \ldots; (P'_n, N'_n))\\
&= (Q,\ ((P_1\vee P'_1), (N_1\vee N'_1));\\
&\qquad (\mathcal{P}_2\wedge(P'_2\vee\mathcal{X})\wedge(P_2\vee\mathcal{X}'),\ \mathcal{N}_2\wedge(N'_2\vee\mathcal{X})\wedge(N_2\vee\mathcal{X}')); \ldots;\\
&\qquad (\mathcal{P}_n\wedge(P'_n\vee\mathcal{X})\wedge(P_n\vee\mathcal{X}'),\ \mathcal{N}_n\wedge(N'_n\vee\mathcal{X})\wedge(N_n\vee\mathcal{X}')))
\end{aligned}$$

with $\mathcal{X} = ((\neg x_1\vee P_1)\wedge(x_1\vee N_1))$, $\mathcal{X}' = ((\neg x_1\vee P'_1)\wedge(x_1\vee N'_1))$ and the recursive call:

$$\begin{aligned}
&(Q', (\mathcal{P}_2, \mathcal{N}_2); \ldots; (\mathcal{P}_n, \mathcal{N}_n)) =\\
&\ (Q', (P_2, N_2); \ldots; (P_n, N_n)) \oplus (Q', (P'_2, N'_2); \ldots; (P'_n, N'_n))
\end{aligned}$$

[1] Since we need one solution, we privilege the interpretation of x to **true**.

This operator is the counterpart of the disjunction for the QBF. In the previous definition when $n = 1$,

$$(q_1 x_1, (P_1, N_1)) \oplus (q_1 x_1, (P'_1, N'_1)) = (q_1 x_1, ((P_1 \vee P'_1), (N_1 \vee N'_1)))$$

which defines the base case of recursivity of $\oplus$. We develop for the case $n = 2$ the disjunction of the matrices of the interpretation of two literal bases and show how we can find back Definition 7: Since $(q_2 x_2, (P_2, N_2)) \oplus (q_2 x_2, (P'_2, N'_2)) = (q_2 x_2, ((P_2 \vee P'_2), (N_2 \vee N'_2)))$, $\mathcal{P}_2 = (P_2 \vee P'_2)$ and $\mathcal{N}_2 = (N_2 \vee N'_2)$ then

$$\begin{aligned}
&((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1)) \wedge ((\neg x_2 \vee P_2) \wedge (x_2 \vee N_2)) \vee \\
&((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1)) \wedge ((\neg x_2 \vee P'_2) \wedge (x_2 \vee N'_2)) \\
&\ \equiv (\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1)) \wedge \\
&\quad (\neg x_2 \vee ((P_2 \vee P'_2) \wedge (P'_2 \vee ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1))) \wedge (P_2 \vee ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1))))) \wedge \\
&\quad (\neg x_2 \vee ((N_2 \vee N'_2) \wedge (N'_2 \vee ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1))) \wedge (N_2 \vee ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1))))) \\
&\ \equiv (\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1)) \wedge \\
&\quad (\neg x_2 \vee (\mathcal{P}_2 \wedge (P'_2 \vee \mathcal{X}) \wedge (P_2 \vee \mathcal{X}'))) \wedge (\neg x_2 \vee (\mathcal{N}_2 \wedge (N'_2 \vee \mathcal{X}) \wedge (N_2 \vee \mathcal{X}')))
\end{aligned}$$

Definition 7 may be improved with no cost by applying, as simplification rules, some usual logical equivalences: $(x \wedge x) \equiv x$, $(x \vee x) \equiv x$, $(x \wedge \neg x) \equiv \bot$ and $(x \vee \neg x) \equiv \top$ with x a propositional variable; $(H \wedge \top) \equiv H$, $(H \wedge \bot) \equiv \bot$, $(H \vee \top) \equiv \top$ and $(H \vee \bot) \equiv H$ with H a propositional formula.

Theorem 6. *Let Q be a binder and $lb, lb' \in \mathcal{B}_Q$ such that $lb^* = QM$ and $lb'^* = QM'$. Then $(lb \oplus lb')^* = QM_\oplus$ with $M_\oplus \equiv (M \vee M')$.*

We are now able to present the search-based algorithm *search_comp_qbf* which compiles a QBF into an optimal literal base. The *search_comp_qbf* algorithm checks first if the binder is reduced to a single quantifier with its variable. If it is the case and if $M \equiv \top$, conversely to *search_certif_qbf* algorithm, $(\top, \top)$ is returned (since $\exists x \top \cong \exists x((\neg x \vee \top) \wedge (x \vee \top))$) in order to compose the two possibilities. If there are some quantifiers, since the algorithm is a search-based one, the first one is considered. Following semantics of QBF, if there is no model for one (resp. both) recursive call then there is no model for the QBF if the quantifier is universal (resp. existential) ; if there are models for both recursive calls then, for both quantifiers, $(Q, (\top, \bot)\ ;\ grds(lb^+)) \oplus (Q, (\bot, \top)\ ;\ grds(lb^-)$ is returned.

Literal bases generated by the *search_comp_qbf* compilation algorithm may be in worst case of exponential size.

Theorem 7 (Correctness of *search_comp_qbf*). *Let QM be a QBF. search_comp_qbf(Q, M) returns a literal base lb such that $lb^* \cong QM$ if QM is valid and returns non_valid otherwise.*

We can now establish that the literal base generated by the *search_comp_qbf* algorithm is optimal.

Theorem 8 (Optimality of *search_comp_qbf*). *Let QM be a valid QBF. Then search_comp_qbf(Q, M) is optimal.*

In case of search-based algorithms for CNF QBF, unit propagation may be easily added ; but, conversely to *search_certif_qbf* algorithm, monotone literal

Algorithm 2. *search_comp_qbf*

In: Q : a binder of a QBF
In: M : a matrix of a QBF
Out: an optimal literal base or *non_valid*

```
if Q = qx then
  if q = ∃ then
    switch M do
      case ⊤ :  return (∃x, (⊤, ⊤))
      case ⊥ :  return non_valid
      case x :  return (∃x, (⊤, ⊥))
      case ¬x :  return (∃x, (⊥, ⊤))
    end switch
  else
    if M = ⊤ then return (∀x, (⊤, ⊤)) else return non_valid end if
  end if
else
  Q = qxQ'
  lb+ := search_comp_qbf(Q', M[x ← ⊤])
  lb− := search_comp_qbf(Q', M[x ← ⊥])
  if q = ∃ then
    if lb+ = non_valid and lb− = non_valid then return non_valid end if
    if lb+ = non_valid then return (Q, (⊥, ⊤) ; grds(lb−)) end if
    if lb− = non_valid then return (Q, (⊤, ⊥) ; grds(lb+)) end if
    return  (Q, (⊤, ⊥) ; grds(lb+)) ⊕ (Q, (⊥, ⊤) ; grds(lb−))
  else
    if lb+ = non_valid or lb− = non_valid then
      return  non_valid
    else
      return  (Q, (⊤, ⊥) ; grds(lb+)) ⊕ (Q, (⊥, ⊤) ; grds(lb−))
    end if
  end if
end if
```

propagation can not be applied to the *search_comp_qbf* algorithm since it does not preserve all the models.

6 Related Work

QBF Certificates. To the best of our knowledge, there exist only two suggestions for QBF certificates and methods to generate them[2]. The first approach [20] is a method to generate a list of pairs of the form (v, f_v) where f_v are the Skolem functions for fresh variables v from the classical extension rule for propositional

[2] The approach proposed in [27] is a method to generate a subset of the clauses of a QBF formula in prenex normal form which is non-valid from traces of search-based solvers. Since this approach is focused on non-validity, it is out of the scope of this paper which is focused on validity.

logic [26]. The second approach proposed in [3,4] introduces the **sat**-certificate. It is described independently of any algorithm, but with binary decision diagrams (BDD) [8] and only for formulae in CNF. The computation of a **sat**-certificate is described in [2] in the framework of sKizzo as a reconstruction from a trace: the "inference log". An external certifier application (ozziKs) is charged with interpreting the content of the log in order to construct certificates [4]. Since the solver can choose between five different inference strategies there are many different kinds of instructions in the inference logs. It results in the need for a heavyweight proof checker. This approach is based on a trace of what the solver is doing and it probably does not scale well because of the growth of this trace. It can take more time to generate the **sat**-certificate from the trace than it took to generate the model [4].

QBF Compilation. Knowledge compilation with a subset of the propositional language as a target language has been widely study (see [14] for a "knowledge compilation map"), but it is not the case for QBF compilation: [16] focuses on selected propositional fragments and quantifier elimination while [12] focuses on complexity of QBF built on the same selected propositional fragments. The compiler for CNF QBF proposed in [25] extends a quantifier-elimination decision procedure [22] as follows: for a CNF QBF $q_1x_1 \dots q_{n-1}x_{n-1}q_nx_nM$, we compute the formulae M_n, P_n and N_n defined on $\{x_1, \dots, x_{n-1}\}$ such that $M \equiv (M_n \wedge ((\neg x_n \vee P_n) \wedge (x_n \vee N_n)))$; if $q_n = \exists$ then the process is recursively called on $q_1x_1 \dots q_{n-1}x_{n-1}(M_n \wedge (P_n \vee N_n))$ otherwise the process is recursively called on $q_1x_1 \dots q_{n-1}x_{n-1}(M_n \wedge (P_n \wedge N_n))$. The target language of this approach is similar to the literal base language: $(q_1x_1 \dots q_nx_n, (P_1, N_1); \dots; (P_n, N_n))$ is a literal base. Since $(P_n \vee N_n)$ is not CNF, the expansion of the existential quantifier for CNF is involved with a quadratic size increase of the formula [21]. Clearly enough the literal base generated by this quantifier-elimination compiler is optimal and it is usually smaller than the literal base generated by the *search_comp_qbf* with out simplifications.

7 Concluding Remarks

We have described in this article a unified framework for **sat**-certificate and compilation of QBF. We have proposed a search-based procedure to compute **sat**-certificates which is very useful since most QBF solvers are search-based decision procedures.

Literal bases generated by the *search_comp_qbf* compilation algorithm may be in worst case of exponential size what complies with complexity results [13]. Anyway, we think that compilation is useful since all the solutions are kept and decision over existentially quantified variables may be not fully described in the QBF. In that case, for each existentially quantified variable, the two different possibilities are computed in polynomial time thanks to optimality and if both substitutions take part of a solution, the choice is left to the user, following its preferences.

Acknowledgement. We would like to thank the referees for their comments which helped improve this paper.

References

1. Ayari, A., Basin, D.: Qubos: Deciding quantified boolean logic using propositional satisfiability solvers. In: Aagaard, M.D., O'Leary, J.W. (eds.) FMCAD 2002. LNCS, vol. 2517, pp. 187–201. Springer, Heidelberg (2002)
2. Benedetti, M.: Evaluating QBFs via Symbolic Skolemization. In: Baader, F., Voronkov, A. (eds.) LPAR 2004. LNCS, vol. 3452, pp. 285–300. Springer, Heidelberg (2005)
3. Benedetti, M.: Extracting certificates from quantified boolean formulas. In: Proceedings of 9th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 47–53 (2005)
4. Benedetti, M.: skizzo: a suite to evaluate and certify QBFs. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS, vol. 3632, pp. 369–376. Springer, Heidelberg (2005)
5. Benedetti, M., Mangassarian, H.: Experience and perspectives in qbf-based formal verification. Journal on Satisfiability, Boolean Modeling and Computation (to appear, 2008)
6. Biere, A.: Resolve and expand. In: Hoos, H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 59–70. Springer, Heidelberg (2005)
7. Bordeaux, L.: Boolean and interval propagation for quantified constraints. In: First International Workshop on Quantification in Constraint Programming (2005)
8. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers 35(8), 677–691 (1986)
9. Büning, H.K., Karpinski, M., Flögel, A.: Resolution for quantified boolean formulas. Information and Computation 117(1), 12–18 (1995)
10. Büning, H.K., Subramani, K., Zhao, X.: Boolean functions as models for quantified boolean formulas. Journal of Automated Reasoning 39(1), 49–75 (2007)
11. Cadoli, M., Schaerf, M., Giovanardi, A., Giovanardi, M.: An algorithm to evaluate quantified boolean formulae and its experimental evaluation. Journal of Automated Reasoning 28(2), 101–142 (2002)
12. Coste-Marquis, S., Le Berre, D., Letombe, F., Marquis, P.: Propositional fragments for knowledge compilation and quantified boolean formulae. In: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005), pp. 288–293 (2005)
13. Coste-Marquis, S., Fargier, H., Lang, J., Le Berre, D., Marquis, P.: Representing policies for quantified boolean formulae. In: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006), pp. 286–296 (2006)
14. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research 17, 229–264 (2002)
15. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communication of the ACM 5 (1962)
16. Fargier, H., Marquis, P.: On the use of partially ordered decision graphs in knowledge compilation and quantified boolean formulae. In: AAAI (2006)
17. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)

18. Giunchiglia, E., Narizzano, M., Tacchella, A.: Clause/term resolution and learning in the evaluation of quantified boolean formulas. Journal of Artificial Intelligence Research 26, 371–416 (2006)
19. Bacchus, F., Samulowitz, H.: Using SAT in QBF. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 578–592. Springer, Heidelberg (2005)
20. Jussila, T., Biere, A., Sinz, C., Kröning, D., Wintersteiger, C.: A first step towards a unified proof checker for QBF. In: Marques-Silva, J., Sakallah, K.A. (eds.) SAT 2007. LNCS, vol. 4501, pp. 201–214. Springer, Heidelberg (2007)
21. Lonsing, F., Biere, A.: Nenofex: Expanding NNF for QBF Solving. In: Kleine Büning, H., Zhao, X. (eds.) SAT 2008. LNCS, vol. SAT 2008, pp. 196–210. Springer, Heidelberg (2008)
22. Pan, G., Vardi, M.Y.: Symbolic Decision Procedures for QBF. In: International Conference on Principles and Practice of Constraint Programming (2004)
23. Plaisted, D.A., Biere, A., Zhu, Y.: A satisfiability procedure for quantified Boolean formulae. Discrete Applied Mathematics 130, 291–328 (2003)
24. Rintanen, J.: Constructing conditional plans by a theorem-prover. Journal of Artificial Intelligence Research 10, 323–352 (1999)
25. Stéphan, I.: Finding models for quantified boolean formulae. In: First International Workshop on Quantification in Constraint Programming (2005)
26. Tseitin, G.S.: On the complexity of derivation in propositional calculus. In: Slisenko, A.O. (ed.) Studies in Constructive Mathematics and Mathematical Logic, Part 2, pp. 115–125. Consultants Bureau, New York (1970)
27. Yu, Y., Malik, S.: Validating the result of a quantified boolean formula (QBF) solver: theory and practice. In: Proceedings of the 2005 conference on Asia South Pacific design automation (ASP-DAC 2005), pp. 1047–1051 (2005)
28. Zhang, L.: Solving QBF with combined conjunctive and disjunctive normal form. In: National Conference on Artificial Intelligence (AAAI 2006) (2006)

Towards Decidability of Conjugacy of Pairs and Triples

Benny George Kenkireth[1] and Samrith Ram[2]

[1] School of Technology and Computer Science, Tata Institute of Fundamental Research, Colaba, Mumbai 400005, India
[2] Department of Mathematics, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

Abstract. The equation $XZ = ZY$ is called the conjugacy equation. Here X, Y and Z are languages over a finite alphabet. Given two sets X and Y, we can ask "Does there exist a Z which makes the conjugacy equation true?". We answer this question partially in the case when one of them is a two element set and the other is a three element set.

1 Introduction

The notion of a *word* is central to computer science and its importance can't be overemphasized. Computer scientists have studied words using an automata theoretic approach. In this approach, the emphasis is more on the properties of a set of words or *language* and automata for recognizing the languages. Descriptive complexity theorists have studied individual words. Chaitin's Ω[1], *Kolakoski word* [2][3] and *Thue-Morse word* [4][5] are examples of some words that have merited an individual study.

In 1977 G. S. Makanin [6] proved that it is decidable whether a *word equation* has a solution. This algorithm was extremely complex to program [7] and had an impractical running time [8]. Recently in 1999 Plandowski [9] gave a much simpler algorithm and also showed that the problem belongs to PSPACE. Although satisfiability of equations over words is decidable, when it comes to equations over languages hardly anything is known. Solutions of even some simple equations are not fully understood. We present here our attempts at one such equation.

We study the problem of *Conjugacy of Languages*. Two sets X and Y are called *conjugates* of each other if there exists a set Z such that $XZ = ZY$. The previous best result in this regard is by Cassaigne et al. [10] [11] and they characterize all sets which are conjugated via a two-element biprefix set Z, two element sets which are conjugates, and for all biprefix codes. We have obtained some partial characterizations in the next higher case namely when one set is a pair and the other is a triple.

This rest of this paper is organized as follows. Section 2 gives the preliminary definitions and proves some basic facts which will be used in the remaining

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 224–231, 2009.

sections. Section 3 gives the main result of this paper, namely the partial characterization of two-three conjugacy. The last section discusses some possible extensions of this work.

2 Preliminaries

Let Σ be a finite alphabet and let Σ^+ denote the free semigroup generated by Σ. Σ^* denotes the free monoid generated by Σ. Concatenation is the basic operation on words and it can be easily extended to sets of words or *languages* as they are called. We will use lowercase letters u, v, w, x, y, z with superscripts and subscripts to denote words over Σ^*. 1 stands for the empty string. Uppercase letters denote languages. We shall use a, b and c when we need to differentiate between variables and constants. If w is a word then $|w|$ denotes the length of w. A language X having the property that every word in it is of equal length is called a $uniform$ language.

Definition 1. *Let $x, y, z \in \Sigma^*$. We say that x and y are conjugates of each other if there exists a z such that $xz = zy$.*

Proposition 1. *Let x, y and z be words. The following statements are then equivalent.*

(i) $xz = zy$.
(ii) There exists u, v and $\alpha \geq 0$ such that $x = (uv), y = (vu)$ *and* $z = (uv)^\alpha u$.

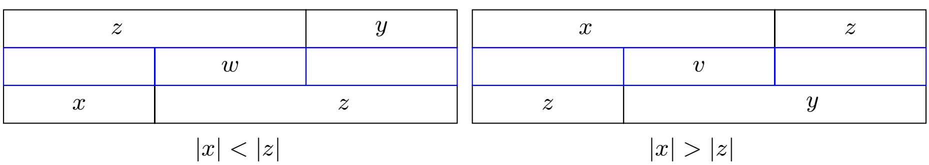


Fig. 1. x and y being conjugates via z

Proof. (ii)⇒(i) is straightforward.

(i)⇒(ii) If $|x| = |z|$ then $x = z = y$ and so we can see that the claim is obviously true. If $|x| > |z|$ we can see from rhs of Figure 1 that $z = u$, $x = uv$ and $y = vu$ satisfies (ii). If $|x| < |z|$ then from lhs of Figure 1, there is a smaller word w such that $xw = z = wy$ and thus makes x and y conjugates. By induction on the length of z, we can now conclude that $x = (uv)$, $w = (uv)^k u$ for some $k \geq 0$ and $y = vu$ which makes $z = (uv)^{k+1}u$. Q.E.D.

Definition 2. *Let x and y be words over Σ . We say that x and y commute iff $xy = yx$.*

By using Proposition 1 and by induction on the length of x we can show that two words commute iff they are powers of the same word. In fact, it is a simple exercise to show that two words commute iff they satisfy a non-trivial equation. We state it as a proposition.

Proposition 2. *Let x and y be words. The following statements are then equivalent.*

(i) $xy = yx$
(ii) $x = p^{r_1},\ y = p^{r_2}$, where $r_1, r_2 \geq 0$
(iii) x *and* y satisfy a non-trivial equation in x and y.

Definition 3. *A non empty word $w \in \Sigma^+$ is called primitive if $w = u^k$ implies that $k = 1$.*

Notice that any word w can be written as p^α in a unique way. Also if w' is a conjugate of w then $w' = q^\alpha$ where q is a conjugate of p. Primitive words are crucial in the study of many word properties. The following proposition characterizes primitive words.

Proposition 3. *Let $p \in \Sigma^+$ be a primitive word. and let $p^2 = xpy$ where x and y are words. Then either $x = 1$ or $y = 1$. In other words a primitive word cannot be a non-trivial factor of its square.*

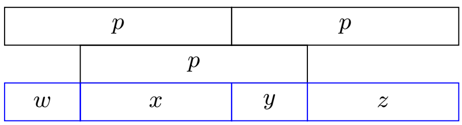


Fig. 2. Primitive word occurring non-trivially

Proof. We will show that the situation as shown in Figure 2 doesn't arise. Assume that the above situation does occur. Length and position consideration forces y to be equal to w and z to be x. Now looking at the first p in p^2 and the p which occurs non trivially, we have $wx = xw$. In other words w and x commute. Thus w and x are powers of the same word. Hence p is also the power of the same word which makes p is a non-primitive word. Thus we have a contradiction. Q.E.D.

Note that using the above characterization, a word is primitive if and only if ww doesn't contain w in a non trivial fashion. This in conjunction with the Knuth-Morris-Pratt algorithm [12] provides a linear time algorithm for checking primitiveness of any word.

Proposition 4. *Let X and Y be languages that are conjugates via Z. Then the number of elements in X and Y of minimal length must be equal.*

Proof. Let X', Y' and Z' be the set of minimal length elements in the sets X, Y and Z respectively. Since the product of X' (resp. Y') and Z' are the minimal length elements in XZ (resp ZY), it must be that $X'Z' = Z'Y'$. Let $|X'| = n_1$, $|Y'| = n_2$ and $|Z'| = n$. Since all elements of X' are Y' are of minimal length (and hence the of the same length) they are biprefix codes. Hence every element obtained by a product of a word in X' and a word in Z' is unique. Thus $X'Z'$ has exactly $n_1.n$ elements. Similarly, $Z'Y'$ has $n.n_2$ elements. Since they must be equal, we have $n_1 = n_2$. Q.E.D.

Proposition 5. *Let X and Y be two languages that are conjugates via Z. Then the elements of X commute if and only if the elements of Y commute. In particular, the elements of X are powers of a primitive word and the elements in Y are the powers of a conjugate word.*

Proof. Without loss of generality, lets assume that the elements of X commute. Hence using Proposition 2, we know that all words of X are of the form w^{r_i} where w is a primitive word. Hence there is a minimal length element in X. Let it be x_1 and of the form w^r for a primitive word w. Using proposition 4, we know that Y also has a minimal length element. Since the minimal length elements have to conjugates of each other we may assume that the smallest element in Y be y_1 and is of the form w'^{r_1} such that w' is a conjugate of w. Note that whenever $XZ = ZY$, we have $X^nZ = X^{n-1}ZY = \cdots = XZY^{n-1} = ZY^n$ i.e. whenever two sets are conjugates, their powers are also conjugates. Let y_i be any word in Y. Since $X^nZ = ZY^n$, we have

$$zy_1y_iy_1^n = x_{i_1} \ldots x_{i_{n+2}}z'$$

But since all the elements of X commute we can write the rhs as $w^{n'}z'$. Similarly from

$$zy_iy_1^{n+1} = x_{i_1} \ldots x_{i_{n+2}}z''$$

we can write the rhs as $w^{n''}z''$. Since by choice of n, we can make n and n' arbitrarily large, the rhs of both the above equations can be made to have matching prefixes. Hence the lhs must also have matching prefixes and hence zy_iy_1 and zy_1y_i being prefixes of the lhs and by virtue of being equal in length must be equal as strings. Cancelling z, we see that y_1 and y_i commute. Thus using Proposition 2 we see that y_i is also a power of w'. Since y_i was an arbitrary word in Y, we can now say that all words in Y are powers of w'. Thus all the elements in Y commute with each other. Q.E.D.

3 Partial Characterization of Two-Three Conjugacy

We will give a partial characterization of two-three conjugacy in this section. The techniques used are combinatorial in nature and are similar to the ones used in Cassaigne et al. [10]. Without loss of generality we may assume that $X = \{x_1, x_2\}$ is the two element set and $Y = \{y_1, y_2, y_3\}$ is the three element set. We may also assume that $|x_1| \leq |x_2|$ and $|y_1| \leq |y_2| \leq |y_3|$. Given two such sets X and Y, we try to answer the question does there exist Z such that $XZ = ZY$. We do a case by case analysis. Note that for such a Z to exist, it must be the case that $|x_1| = |y_1|$. Otherwise the smallest element of XZ will be of a different size from the smallest element in ZY.

An exhaustive list of various cases under the above assumptions is listed here.

1. $|x_1| = |x_2|$
2. $|x_1| < |x_2|$
   i) $|x_2| > |y_3|$
   ii) $|x_2| < |y_2|$

iii) $|y_2| < |x_2| < |y_3|$
iv) $|y_2| < |x_2| = |y_3|$
v) $|y_2| = |x_2| \leq |y_3|$

We give full characterizations of two-three conjugacy for all the above cases except 2(iv) and 2(v). Note that solving these cases will provide us with a decision procedure for two-three conjugacy, that is given any two sets X and Y, one being of size two and the other of size three, we will be able to tell whether these sets are conjugates.

3.1 Characterization Theorems

We provide the characterization by a series of propositions.

Deciding Conjugacy When $|x_1| = |x_2|$

Proposition 6. *Let $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2, y_3\}$ be subsets of Σ^* such that $XZ = ZY$ for some $Z \subseteq \Sigma^*$. Also let $|x_1| = |x_2|$. Then the following conditions hold true.*

(i) $|y_1| = |y_2|$
(ii) ($X = \{pu, pv\}$ *and* $Y = \{up, vp, (up + vp)^{l_1}\}$) or
($X = \{up, vp\}$ *and* $Y = \{pu, pv, (pu + pv)^{l_2}\}$) for some l_1 or l_2.

By $(up + vp)^l$, we mean any word made up of up's and vp's, l of them in all.

Proof. Let Z_1 and Y_1 be words of Z and Y respectively having minimal lengths.

(i) From Proposition 4, we have that the minimal length elements X and Y must be of equal number. Since there are two elements in X of minimal length, there must be two such in Y also. Thus proved.
(ii) From above we have $|x_1| = |x_2| = |y_1| = |y_2|$. Also considering the minimal length elements in XZ and ZY, we know that X and Y_1 are conjugates. Let $Z^{(i)}$ be the set of words from Z having length i. Note $Z^{(i)}Y_1$ is a subset of XZ. Further, since $X$and Y_1 are uniform languages of equal size, $Z^{(i)}Y_1$ must be equal to $XZ^{(i)}$. Since Z is a disjoint union of $Z^{(i)}$'s we have $XZ = ZY_1$. Therefore, now we have two element sets X and Y_1 which are conjugates via Z. Thus, from[3] Cassaigne et al. [10], we know there exist words p, u, v and a set $I \subseteq \mathbb{N}$ such that $|u| = |v|$ and one of the following conditions hold true.
(a) $X = \{pu, pv\}, Y_1 = \{up, vp\}$ and $Z = \bigcup_{i \in I} \{pu, pv\}^i p$
(b) $X = \{up, vp\}, Y_1 = \{pu, pv\}$ and $Z = \bigcup_{i \in I} \{up, vp\}^i \{u, v\}$

[3] Interested reader may consult Lemma 4.5 from the paper referred to.

What remains to be shown is that y_3 is of the form $(up+uv)^l$ or $(pu+pv)^l$ i.e. $y_3 \in \{y_1, y_2\}^*$. We will show this for case (a). The other situation can be argued on similar lines. Let $|x_1|$ be n and $|p|$ be m. For any $z \in Z$ we have $|z| = k \times n + m$ for some non negative k. Consider the equation $zy_3 = x_i z'$. Taking length on both sides and simplifying yields $|y_3| = k'n$ for some positive k'. Also we have $x_i z'$ to be of the form $(pu+pv)^i p$. Taking into consideration the length of y_3 and the fact that y_3 is a suffix of $x_i z'$, we have $y_3 = (up+vp)^{k'}$ Q.E.D.

Observe that when the shapes of X and Y as described in the proposition,one can easily construct Z which makes X and Y conjugates. Hence the proposition gives a characterization for the case when $|x_1| = |x_2|$.

Deciding Conjugacy When $|x_1| < |x_2|$

Proposition 7. *Let $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2, y_3\}$ be sets such that $|x_1| <$ $|x_2|$ $x_2| > |y_3|$ and $XZ = ZY$ for some Z. Then, there exists conjugate words w and w' such that all elements in X are the powers of w and all words in Y are powers of w'.*

Proof. Let z_1 be a smallest word in Z. Since $ZY^{n+1} = X^{n+1}Z$ for any n, We have $z_1 y_3 y_1^n = x_{i_1} \ldots x_{i_{n+1}} z'$ for some $z' \in Z$. Since the lengths of both the sides are equal we have $|z_1| + |y_3| + n|y_1| = |x_{i_1}| + \ldots + |x_{i_{n+1}}| + |z'|$. If any of the x_{i_j} were x_2, since $|x_2| > |y_3|$ then all the x_j put together have length greater than all the y_j put together forcing z' to be smaller than z which is a contradiction. So all x_i's are x_1 and thus

$$z_1 y_3 y_1 y_1^{n-1} = x_1^{n+1} z'$$

Using a similar argument we can show that

$$z_1 y_1 y_3 y_1^{n-1} = x_1^{n+1} z''$$

Looking at the prefixes of both the rhs, we know that they can be made to match for arbitrary length by choosing n appropriately. But since $z_1 y_1 y_3$ and $z_1 y_3 y_1$ are prefixes of the lhs and since they are of equal length, they must now be equal. Thus y_1 and y_3 commute. Also since $|x_2| > |y_3|$ we also have $|x_2| > |y_2|$. Using y_2 instead of y_3 in the above argument we get that y_2 also commutes with y_1. Thus elements of Y commute. Now, using Proposition 5 it is easy to see that proposition is true. Q.E.D.

Proposition 8. *Let $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2, y_3\}$ be sets such that $|x_1| <$ $|x_2|, |y_2| > |x_2|$ and $XZ = ZY$ for some Z. Then, there exists conjugate words w and w' such that all elements in X are the powers of w and all words in Y are powers of w'.*

We skip the detailed proof of this because this proof is exactly similar in spirit to that of Proposition 7. We take a suitable word in $X^n Z$ and use this to show that X commutes and hence Y too commutes. Observe that in the previous proof we were using the fact that elements of Y commuteto prove that elements of X must also commute.

Proposition 9. *Let $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2, y_3\}$ be sets such that $|x_1| < |x_2|$, $|y_2| < |x_2| < |y_3|$ and $XZ = ZY$ for some Z. Then, there exist conjugate words w and w' such that all elements in X are the powers of w and all words in Y are powers of w'.*

Proof. By arguments similar to the ones in the proofs above, we can show that since $|y_2| < |x_2|$, $|y_2|$ and $|y_1|$ must commute. Let z_1 be the smallest word in Z. Take the words $w_1 = x_1^{n-1}x_2x_1z_1$ and $w_2 = x_1^n x_2 z_1$. They both must be of the shape $z'y_{i_1} \dots y_{i_{n+1}}$. Since x_2 is smaller than y_3 by length, none of the y_i's are y_3. But since y_1 and y_2 commute they must be the powers of the same word. That will make w_1 and w_2 to have the same suffix. Hence elements of X commute. Since elements of X commute so must elements of Y. Q.E.D.

In the conditions given by Propositions 7, 8 and 9 we obtain a necessary condition for conjugacy. But note that these are sufficient conditions as well because under these conditions, the elements in X are powers of a primitive word of the form ab and elements in Y are powers of ba. We also know that the minimal elements in these sets of of same length. Any two such sets can be made conjugates via $Z = (ab)^*a$.

We combine the Propositions 7, 8, 9 and 6 to present the main theorem.

Theorem 1. *Let $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2, y_3\}$ be two subsets of Σ^*. If the elements in X are of equal length, X and Y are conjugates if and only if they are of the form given by equation (1) or (2). When they are not of equal length, if any of the conditions given by (3), (4) and (5) are satisfied,X and Y are conjugates if and only if there exists conjugate words w and w' such that words in X are powers of w and words in Y are powers of w'.*

$$|y_1| = |y_2|, \ X = \{pu, pv\} \text{ and } Y = \{up, vp, (up + vp)^{l_1}\} \tag{1}$$

$$|y_1| = |y_2|, \ (X = \{pu, pv\} \text{ and } Y = \{up, vp, (up + vp)^{l_1}\} \tag{2}$$

$$|x_2| > |y_3| \tag{3}$$

$$|y_2| > |x_2| \tag{4}$$

$$|y_2| < |x_2| < |y_3| \tag{5}$$

Proof. For the 'only if' case when $|x_1| = |x_2|$ follows from Proposition 6. The other part follows from Proposition 7, 8 and 9. The 'if' direction is easy to verify and has been mentioned along with the corresponding propositions for the 'only if' direction. Q.E.D.

4 Conclusion

Although two-three conjugacy appears to be a simple case of an easily stated word problem, its full characterization doesn't seem very straightforward. Also this seems to suggest that as the number of elements in X and Y increase, the corresponding conjugacy problems will get correspondingly harder. The characterization given here can easily be seen to be polynomial time algorithm. The

decidability of the general conjugacy problem is still open and does not seem to be amenable to the techniques used here. But we do believe that using techniques described here will be sufficient to characterize two-three conjugacy fully.

References

1. Chaitin, G.J.: Meta Math!: The Quest for Omega. Vintage Books (2006)
2. Lepistö, A.: Repetitions in Kolakoski sequence. In: Rozenberg, G., Salomaa, A. (eds.) Developments in Language Theory, pp. 130–143. WS (1994)
3. Kolakoski, W.G.: Elementary problem 5304. AMM 72, 674 (1965); Solution 73, 681–682 (1966)
4. Berstel, J.: Axel Thue's Papers on Repetitions in Words: a Translation. Publications du Laboratoire de Combinatoire et d'Informatique Mathématique. Université du Québec à Montréal (February 1995)
5. Thue, A.: Über unendliche Zeichenreihen. Norske vid. Selsk. Skr. Mat. Nat. Kl 7, 1–22 (1906); Thue, A., Nagell, T. (eds.) Selected Mathematical Papers, Universitetsforlaget, Oslo, pp. 139–158 (reprinted, 1977)
6. Makanin, G.S.: The problem of solvability of equations in a free semi-group. Math. USSR Sbornik 32(2), 129–198 (1977)
7. Abdulrab, H.: Implementation of makanin's algorithm. In: Schulz, K.U. (ed.) IWWERT 1990. LNCS, vol. 572, pp. 61–84. Springer, Heidelberg (1992)
8. Koscielski, A., Pacholski, L.: Complexity of makanin's algorithm. J. ACM 43(4), 670–684 (1996)
9. Plandowski, W.: Satisfiability of word equations with constants is in pspace. J. ACM 51(3), 483–496 (2004)
10. Cassaigne, J., Karhumäki, J., Manuch, J.: On conjugacy of languages. ITA 35(6), 535–550 (2001)
11. Cassaigne, J., Karhumäki, J., Salmela, P.: Conjugacy of finite biprefix codes. In: Kunc, M., Okhotin, A. (eds.) Theory and Applications of Language Equations, Proceedings of the 1st International Workshop, July 2007, vol. 44, pp. 33–42. TUCS General Publication, Turku Centre for Computer Science (2007)
12. Knuth, D.E., Jr., Morris J.H., Pratt, V.R.: Fast pattern matching in strings. SIAM J. Comput. 6(2), 323–350 (1977)

Gautama – Ontology Editor Based on Nyaya Logic

G.S. Mahalakshmi, T.V. Geetha, Arun Kumar, Dinesh Kumar, and S. Manikandan

Department of Computer Science and Engineering,
Anna University, Chennai, Tamil Nadu, India
{mahalakshmi,tvgeedir}@cs.annauniv.edu,
{arunk.frau2004,dd.jack,write2manikandan}@gmail.com

Abstract. Indian logic based approach of knowledge representation fundamentally classifies the world knowledge into concepts, and relations, both enriched with special qualities. To be more precise, Nyaya Sastra recommends a special categorization of world knowledge which is supposed to be elaborate in tapping the minute details in the defined knowledge units. *Nyaya logics* are a mechanism which defines the concept and relation elements of ontology based on the epistemology of Nyaya-Vaisheshika school of Indian logic. We have already proposed an ontology reference model based on *Nyaya logic*, known as NORM. To develop an ontology using *Nyaya logics*, one should be aware of the syntax and semantics of NORM *rdf*. To overcome the difficulty involved in creating NORM based ontology, in this paper, we propose *Gautama*, a tool for editing the ontology based on *Nyaya logics*. We also discuss the steps for building the ontology for a sample from 'Birds' domain.

Keywords: Indian logic, Nyaya Sastra, NORM, Ontology.

1 Introduction

The Nyaya-Vaisheshika is a self-contained system of philosophy. It proposes a unique categorisation of world knowledge elements [6,9]. Through the epistemological definitions of Nyaya-Vaisheshika system, the treatment of world knowledge elements was very special which contributed to the uniqueness of ontological categorization. The methodology of categorization was inaugurated by Gautama-Akshapada, which consists in enumeration and classification of world knowledge entities into specific categories which were recommended, argued and analysed by the followers of Gautama [6,9].

NORM is the Nyaya based Ontology Reference Model, which defines the standards for constructing ontology, based on the recommendations of the epistemology definitions of Nyaya-Vaisheshika school of Indian philosophy. NORM is organized as a two-layer ontology [8], where the upper layer represents the abstract fundamental knowledge and the lower layer represents the domain knowledge. According to NORM, a node in the ontology is composed of an enriched concept which is related implicitly to its member qualities and explicitly to other peer concepts, by means of relations [11].

A node of Nyaya-Vaisheshika [5,10] based ontology has the following structure (refer Fig. 1). Every concept of the world knowledge shall be thoroughly classified as per NORM structure. The abstract and domain concepts form a strict classification hierarchy. The traits of the top-level concepts are applicable down the hierarchy.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 232–242, 2009.

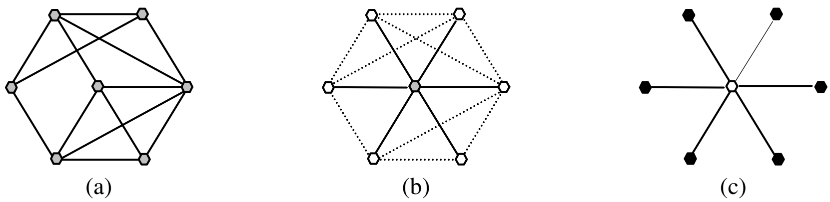


Fig. 1. NORM model for cognitive knowledge representation (a) ontology with concepts as nodes and external relations as edges (b) a concept with qualities as nodes, internal relations as thin edges, tangential relations as dotted edges (c) a quality with values as nodes, grouping relations as edges [8]

Every concept in the NORM model has links to other concepts by external relations (Fig. 1a). A concept is made of qualities or *gunas* [5,10]. In addition, the qualities are bounded to the concept by internal relations. The qualities may also be related to each other, which is depicted as dotted edges (refer Fig. 1.b). Every quality has a set of values. Every value is the substratum of the quality to which it is associated [5,10]. The values are bounded to the qualities by grouping relations (refer Fig. 1c). This model (Fig. 1) is inspired by the various recommendations of classifications of world knowledge according to Nyaya-Vaisheshika. The following section discusses the system of classification of Nyaya Sastra.

2 Nyaya-Vaisheshika System of Classification

According to Nyaya Sastra [4,5,10], every concept is classified into seven categories: substance, quality, action, generality, particularity, inherence and negation. Among these, the substance is of nine kinds: earth, water, light, air, ether, time, space, soul and mind. Every substance is threefold: body, organ and object. The object of light is fourfold: earthly, heavenly, gastric and mineral. Every substance is said to possess some quality. The quality is of twenty-four varieties which in turn possess values (refer Fig. 2).

The permissible action associated with the substance is of five types: upward motion, downward motion, contraction, expansion, and motion. Generality is either more comprehensive or less comprehensive. Particularities are innumerable [4,5,10]. Negation is of four varieties: antecedent negation (or prior negation, destructive negation (or posterior negation, [1]), absolute negation and mutual negation. Out of the nine substances, odour persists only in earth and is inherent. Earth exists in all the seven colors. Air has no color; water is pale-white in color and light is bright-white in color. Air has touch. Water has cold-touch and light has hot-touch. Dimension (or magnitude), distinctness, conjunction and disjunction are present in all the nine substances. Remoteness and Proximity is found in earth, water, light, air and mind. Heaviness or Weight is only in earth and water. Viscidity is present only in the substance, Water [4,5,10].

The detailed structure of a node in Nyaya-Vaisheshika ontology is shown in Fig. 3. The structure incorporates almost all the recommendations of Nyaya-Vaisheshika school along with the detailed definitions of relations at every level, between concepts, between concept and member qualities, between qualities, and between quality and member values. The following section describes the ontology editor, Gautama for editing the world knowledge in the form of Indian logic ontologies.

Color: white, blue, yellow, red, green, brown, varied
Taste: sweet, acid, saline, pungent, astringent, bitter
Odour: fragrant, foul
Touch: cool, hot, lukewarm
Number
Magnitude: atomic, large, long, short
Separateness
Conjunction
Disjunction
Remoteness: spatial, temporal
Proximity: spatial, temporal
Weight
Fluidity: natural, artificial
Viscidity
Sound: articulate, inarticulate
Intellect
Pleasure: ***Remembrance, apprehension***: *Erroneous apprehension, valid apprehension*:
Valid apprehension : perception, inference, analytical knowledge and verbal testimony
Pain
Desire
Aversion
Volition
Merit
Demerit
Tendency

Fig. 2. Ontological Classification of Nyaya-Vaisheshika Qualities

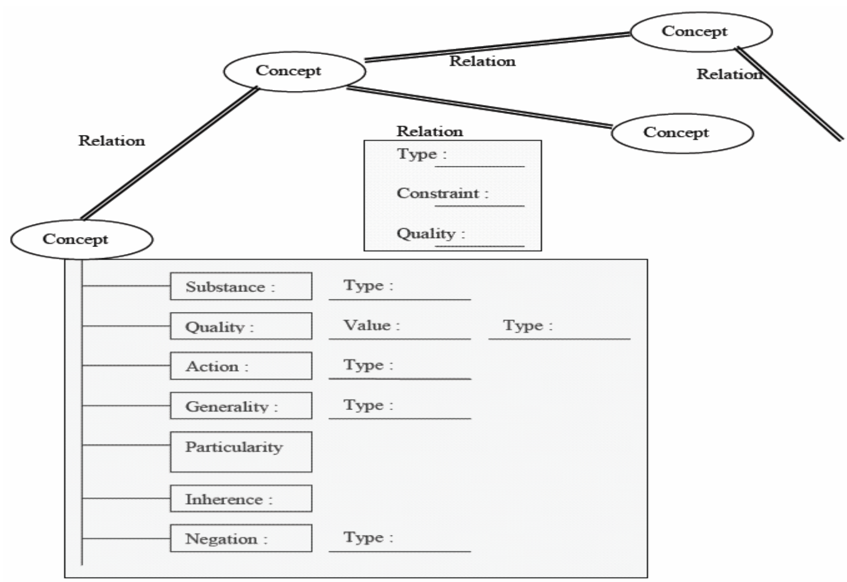


Fig. 3. The node ontology architecture of NORM

3 Gautama – Ontology Editor for Indian Logic

We have developed an ontology editor (refer Fig. 4) known as 'Gautama' for imparting the knowledge in the form of Indian logic. The editor has icons and toolboxes to create / edit the knowledgebase defined under the Nyaya-Vaisheshika system of classification [4,5,10].

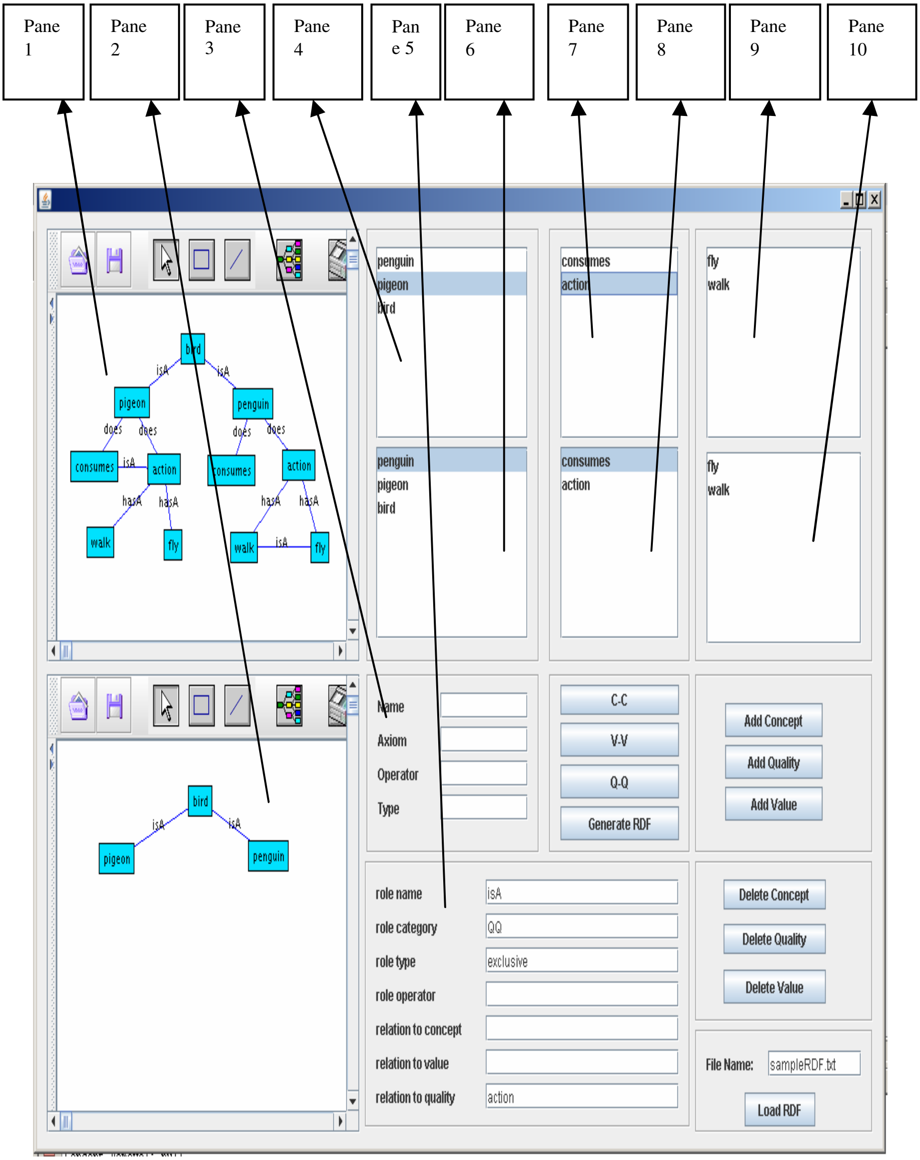


Fig. 4. Gautama – NORM based Ontology editor

The ontology editor has various panes for editing the concepts, qualities and relations, both graphically and through entry forms. Each one of those panes are described as follows:

ILO Visualisation Pane: This pane contains icons to save and print the ontology visualisation created in the top left pane of the editor. In addition, drawing icons have also been provided.

Concepts Visualisation Pane: This pane is similar to ILO Visualisation Pane, except that, here, only the concept hierarchy in the ontology is visualised.

Nodes Entry Pane: This pane provides controls for entering information about the nodes that are yet to be created to become part of the ontology. C-C denotes concept-concept; V-V denotes value-value and Q-Q denotes quality-quality. There are enough command buttons to add concepts, qualities and values. Using these, the concept definitions shall be created. The 'Generate RDF' button helps in generation of Resource description format of the underlying ontology.

Relations Entry Pane: The purpose of this pane is identical to Nodes entry Pane. Here, 'roles' shall be created as part of the ontology. NORM recommends various relations (refer Fig.1), therefore, this pane has provisions for creating relations at all levels. To the extreme right, is the command buttons for 'deletion' services. Using these buttons, concept / quality / value shall be deleted from the ontology. Alternatively, one can also load a pre-existing RDF through 'load rdf' button to have the ontology loaded into the memory at once.

Concepts list Pane: This pane lists all the concepts available in the ontology with specialised concepts first displayed, followed by the generalised concepts. (Please note from Fig. 4.4. that, 'penguin' and 'pigeon' are displayed before 'bird'. There are two concepts list pane, primary and secondary.

Quality List Pane: This pane lists all the qualities available for the selected concept in the adjacent left pane. This is divided into primary and secondary panes.

Value list pane: This pane lists all the values available for the selected quality in the adjacent left quality list pane. This is divided into Primary and secondary panes.

If two concepts are related to each other, one concept and its member qualities, member values shall be seen in the primary pane. Simultaneously, the other concept and its member qualities, member values shall be seen in the secondary pane. To facilitate the recording of knowledge in RDF (resource description format), appropriate tags have been defined, with a start tag and corresponding end tag with the item described in between. The following are the various tags defined for the RDF of Indian logic ontology generated by Gautama.

- <rdf:concept> - This tag is used to declare a concept prior and after its definition.
- <rdf:name> - This tag is used to declare the name of a concept / quality / relation.
- <rdf:desc> - This tag is used to create descriptions or definitions for a particular concept.
- <rdf:axiom> - This tag is used to create concept axioms.
- <rdf:quality> - This tag is used to create member qualities for a given concept.
- <rdf:type> - This tag is used to declare the type of a concept / quality / relation.
- <rdf:role> - This tag is used to declare the role of a concept / quality.
- <rdf:category> - This tag is used to declare the category of relation like external, internal, tangential or grouping.
- <rdf:operator> - This tag is used to declare the logical operators like and, or while creating the concept axioms of the ontology.

The sample RDF generated for a simple ontology for ‘birds’ domain is given in Fig. 5. The facilities for interacting with the knowledgebase are generally done through knowledge representation languages. *NORM model* for knowledge representation involves *Nyaya Description language (NDL)*, the set of commands used for defining the units of knowledge base.

```
<rdf:concept>
        <rdf:name>bird</rdf:name>
</rdf:concept>

<rdf:concept>
        <rdf:name>pigeon</rdf:name>
        <rdf:axiom>bird</rdf:axiom>
        <rdf:desc>
                <rdf:quality>
                        <rdf:name>action</rdf:name>
                        <rdf:type>exclusive</rdf:type>
                        <rdf:operator>and</rdf:operator>
                        <rdf:value>
                                <rdf:name>walk</rdf:name>
                                <rdf:operator>and</rdf:operator>
                        </rdf:value>
                        <rdf:value>
                                <rdf:name>fly</rdf:name>
                                <rdf:operator>inclusive</rdf:operator>
                                <rdf:role>
                                        <rdf:name>isa</rdf:name>
                                        <rdf:category>VVrelationship</rdf:cate
                                        <rdf:type>symmetric</rdf:type>
                                        <rdf:relToValue>walk</rdf:relToValue
                                        <rdf:operator>and</rdf:operator>
                                </rdf:role>
                        </rdf:value>
                        <rdf:role>
                                <rdf:name>hasA</rdf:name>
                                <rdf:category>QVrelationship</rdf:category>
                                <rdf:type>transitive</rdf:type>
                                <rdf:relToValue>walk</rdf:relToValue>
                                <rdf:operator>or</rdf:operator>
                        </rdf:role>
                        <rdf:role>
                                <rdf:name>hasA</rdf:name>
                                <rdf:category>QVrelationship</rdf:category>
                                <rdf:type>transitive</rdf:type>
                                <rdf:relToValue>fly</rdf:relToValue>
                                <rdf:operator>and</rdf:operator>
                        </rdf:role>
                        <rdf:role>
                                <rdf:name>isA</rdf:name>
                                <rdf:category>QCrelationship</rdf:category>
                                <rdf:type>transitive</rdf:type>
                                <rdf:relToConcept>pigeon</rdf:relToConcept>
```

Fig. 5. NORM RDF– ‘penguin’ and ‘pigeon’ example

```
                <rdf:operator>and</rdf:operator>
            </rdf:role>
        </rdf:quality>
        <rdf:quality>
            <rdf:name>consumes</rdf:name>
            <rdf:operator>and</rdf:operator>
            <rdf:type>exclusive</rdf:type>
            <rdf:role>
                <rdf:name>isA</rdf:name>
                <rdf:category>QQrelationship</rdf:category>
                <rdf:type>transitive</rdf:type>
                <rdf:relToQuality>action</rdf:relToQuality>
                <rdf:operator>and</rdf:operator>
            </rdf:role>
            <rdf:role>
                <rdf:name>isA</rdf:name>
                <rdf:category>QCrelationship</rdf:category>
                <rdf:type>transitive</rdf:type>
                <rdf:relToConcept>pigeon</rdf:relToConcept>
                <rdf:operator>and</rdf:operator>
            </rdf:role>
        </rdf:quality>
        <rdf:role>
            <rdf:name>isA</rdf:name>
            <rdf:category>CCrelationship</rdf:category>
            <rdf:type>transitive</rdf:type>
            <rdf:relToConcept>bird</rdf:relToConcept>
            <rdf:operator>and</rdf:operator>
        </rdf:role>
    </rdf:desc>
</rdf:concept>
```

Fig. 5. (*continued*)

The knowledge representation language [1,7], is classified into concept/relationship definition language (CRDL), concept/relation manipulation language (CRML) and a set of editing commands and a query language. This knowledge representation language can be further used to define, manipulate and query the various levels of knowledge. CN refers to Concept name, QN refers to Quality Name, V – Quality value (Ex: color – Indigo: quality: color, value: Indigo) RN refers to Role name, I refer to Instance and Rdesc refers to Role descriptions. The CRDL constitutes the commands for defining the concepts, instances and relationships. Top and Bottom concepts are assumed by the system as default. The concept definitions have been recognized and the knowledge hierarchy is built. Therefore, using CRDL, the user can build the knowledge base right from scratch. Concepts can be linked to one another through relations where relations can be is-a, owns, part-of and uses. Relations and actions can also be defined between concept and quality. Instances of concepts can also be defined using CRDL. Following the above norms of definition of knowledge representation languages (as description logic commands), here, we define the sample Nyaya logic commands which are listed in Table 1.

Table 1. Commands for querying with Gautama

CRDL	**CRML**
define-concept<*CN, Level*> define-concept-axiom<*CN,Cdesc*> disjoint-concept<*C1,C2*> define-role-axiom<*RN, Rdesc*> disjoint-role<*R1,R2*> define-concept-role<*RN,C1,C2*> define-concept-qualities<*CN, (QM,Qman.List) / (QO,Qopt.List) / (QE,Qexceptional.List) / (QX,Qexclusive.List)*> define-quality-values<*CN,QN,V1….Vn*> define-role-quality <*RN,CN, Qreflexive.List / Qsymmetric.List / Qassymmetric.List / Qantisymmetric.List / Qtransitive.List / Qdirect.List / Qindirect.List / Qexclusive.List*> define-quality-role<*RNreflexive.List / RNasymmetric.List / RNsymmetric.List / RNantisymmetric.List / RNtransitive.List / RNdirect.List / RNindirect.List / RNexclusive.List, CN,QN*>	insert-quality<*QN*> delete-quality<*QN*> insert-values<*QN,V1….Vn*> delete-values<*QN,V1….Vn*> delete-concept<*CN*> delete-instance<*I*> update-instance<*I,Cnold,Cnnew*> delete-role-filler<*I1,I2,RN*> update-role-filler<*I1,I2,Rnold,Rnnew*> delete-role<*RN*> insert-role<*RN*> delete-concept-quality<*CN,QN*> delete-quality-value<*CN,QN,VInvariableConcomitance.List / VExclusive.List / VInvariableConcomitance.List / VDirect.List*> insert-quality-value<*CN,QN,VInvariableConcomitance .List / VExclusive.List / VInvariableConcomitance.List / VDirect.List*> update-quality-value<*CN,QN,Vold,Vnew*>
Query language	**Query language**
concept-satisfiable<*CN*> concept-subsumes<*C1,C2*> concept-disjoint<*C1,C2*> chk-concept<*CN*> concept-atomic<*CN*> concept-ancestors<*CN*> concept-descendants<*CN*> super-concept<*CN*> sub-concept<*CN*> chk-concept-related<*C1,C2*> chk-concept-related<*C1,C2,RN*> chk-concept-related<*C1,C2,RNreflexive.List / RNasymmetric.List / RNsymmetric.List / RNantisymmetric.List / RNtransitive.List / RNdirect.List, RNindirect.List / RNexclusive.List* > chk-quality<*QN*> chk-concept-quality<*CN,QN*> all-qualities	retrieve-direct-concepts<*I*> retrieve-indirect-concepts<*I*> retrieve-concept-fillers<*RN,C1*> all-concepts<*I*> retrieve-qualities<*CN*> retrieve-quality-value<*CN,QInvariableConcomitance / QExclusive / QExceptional*> retrieve-quality-value<*CN,QDirect*> chk-instance<*I*> chk-instance-type<*I,CN*> chk-instance-related<*I1,I2*> retrieve-direct-instances<*I*> retrieve-indirect-instances<*I*> retrieve-instance-fillers<*RN,I1*> all-instances<*CN*> retrieve-related-instances<*RN*> *retrieve-quality-value<*I,QN*> chk-role<*RN*> all-roles role-descendants<*CN*> role-ancestors<*CN*>

The CRML provides necessary commands for deleting and updating of concepts and associated relations in the knowledge hierarchy. The query language supports querying the classification hierarchy and to summarize the results of queries. The TAML commands have been utilized for the management of Tbox and Abox. The system shell is managed by create and use taxonomy which are used primarily for mounting and dismounting the Tbox and Aboxes. Upon commit, the information contained in the classification hierarchy is stored in a separate file, which also records every inferencing performed by the system. In addition the system provides concept and instance dictionary files, which summarises the total number of knowledge units present in the classification hierarchy. Using CRML, the ontology shall be updated or modified. The concepts and the relation between concepts can be manipulated using the commands of CRML.

The CRDL and CRML commands are used only during the creation of ontology by end users. To be more user-friendly, 'Gautama', the ontology editor provides built-in facilities for ontology creation and updation services. The query language shall be used with the RDF generated by Gautama, to query about various parts of the ontology. Here, we discuss few commands of the query services.

- Concept-satisfiable – This takes a concept name as the parameter and checks whether the addition of the concept will not violate the ontology definitions that exist prior to the execution of this command.
- Concept-subsumes – This takes two concepts as input, and checks whether the first concept subsumes the second concept. This is one of the famous reasoning service provided by any ontology-based reasoner.
- Concept ancestors and Concept-descendants – These commands list the ancestral / descending concepts in the ontology hierarchy. Role-ancestors and Role-descendants also have similar purpose.
- Sub-concept, Super-concept – These commands retrieve the child nodes or parent nodes of the parametric concept from the ontology hierarchy.
- Chk-concept-related – This command has three variations. It either checks whether a concept is related to another concept, through a particular relation name or through a particular set of relation categories.
- Chk-quality – This command checks the entire ontology hierarchy to check if the required quality is available in the ontology.
- Chk-concept-quality – This command checks the entire ontology hierarchy to check if the particular concept has the required quality.
- All-concepts, all-qualities, all-roles, all-instances – These commands just lists all the concepts, qualities, roles or instances available in the ontology.
- Retrieve-direct-concepts, retrieve-indirect-concepts – The first commands take an instance as input, and retrieve all the directly related concepts to those instances; The second command take the instance as input and retrieves all the second and higher degree concepts related to those instances. For example, if 'TooToo' is an instance of penguin, the first command may retrieve 'penguin' as the result; the second command will retrieve all the ancestors of penguin which are conceptually related to penguin. Retrieve-direct-instances, retrieve-indirect-instances also serve the same purpose.

4 Related Work

This paper proposed the ontology editor based on Indian logic based knowledge representation system. Using this editor, one can carefully handcraft the ontology based on Indian logic in the required domain. However, there are other noteworthy projects existing in the knowledge representation arena. Cyc, WordNet, Concept-Net and Mind-Net are to name a few.

Cyc is an artificial intelligence project [3] that attempts to assemble a comprehensive ontology and database of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning. The Cyc system is made up of three distinct components, all of which are crucial to the machine learning process: the knowledge base (KB), the inference engine, and the natural language system. The Cyc inference engine is responsible for using information in the KB to determine the truth of a sentence and, if necessary, find provably correct variable bindings. The natural language component of the system consists of a lexicon, and parsing and generation subsystems. The lexicon is a component of the knowledge base that maps words and phrases to Cyc concepts.

WordNet is a large lexical database [2] of English, where, Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet's structure makes it a useful tool for computational linguistics and natural language processing. Its design is inspired by current psycholinguistic and computational theories of human lexical memory.

ConceptNet is a freely available [13] commonsense knowledge base and natural-language-processing toolkit built at MIT. The ConceptNet knowledge base is a semantic network of commonsense knowledge encompassing the spatial, physical, social, temporal, and psychological aspects of everyday life. Whereas similar large-scale semantic knowledgebases like Cyc and WordNet are carefully handcrafted, ConceptNet is generated automatically from World Wide Web.

ConceptNet is a unique resource in that it captures a wide range of commonsense concepts and relations, yet this knowledge is structured as a simple, easy-to-use semantic network, like WordNet. While ConceptNet still supports query expansion and determining semantic similarity, its focus on concepts-rather-than-words, its more diverse relational ontology, and its emphasis on informal conceptual-connectedness over formal linguistic-rigor allow it to go beyond WordNet to make practical, context-oriented, commonsense inferences over real-world texts.

A MindNet is a collection of semantic relations that is automatically extracted from text data using a broad coverage parser [12]. MindNets are produced by a fully automatic process that takes the input text, sentence-breaks it, parses each sentence to build a semantic dependency graph (Logical Form), aggregates these individual graphs into a single large graph, and then assigns probabilistic weights to subgraphs based on their frequency in the corpus as a whole. The project also encompasses a number of mechanisms for searching, sorting, and measuring the similarity of paths in a MindNet.

'Gautama' proposed in this paper is not automatic, i.e. it does not harvest ontological entities automatically from the text corpora or web, instead, it is a first step in the

design of an ontology editor based on Indian logic, and therefore, presently it is only handcrafted to serve the purpose. In future, adapting more ideas of building the ontology from Indian philosophy would strengthen the outcome of the ontology editor.

5 Conclusion

This paper proposed the overview of Gautama, a tool for editing the world knowledge elements into ontology based on Indian logic. The ontology followed the guidelines of NORM (Nyaya Ontology reference model) based ontological standards which is built on the epistemological recommendations of Nyaya-Vaisheshika school of Indian philosophy, for defining the knowledge units of the ontology. We hope, this tool, facilitates easy creation of Indian logic based ontologies and thereby promotes the wide study of Indian logic in the ever green field of ontological and philosophical research.

References

1. Aghila, G., Mahalakshmi, G.S., Geetha, T.V.: KRIL – A Knowledge Representation System based on Nyaya Shastra using Extended Description Logics. VIVEK journal 15(3), 3–18 (2003)
2. Fellbaum, C.: WordNet - An Electronic Lexical Database, with a preface by George Miller (May 1998)
3. Matuszek, C., Witbrock, M., Kahlert, R.C., Cabral, J., Schneider, D., Shah, P., Lenat, D.: Searching for Common Sense: Populating Cyc from the Web. In: Proceedings of the Twentieth National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania (July 2005)
4. Vidyabhusana, S.C.: Gautama, The Nyaya Sutras. In: Sinha, N.L. (ed.) Sacred Book of the Hindus, Allahabad (1930); Reprinted in Motilal Banarsidass, Delhi (1990)
5. Ballantyne, J.R.: Lectures on Nyaya Philosophy-Embracing the text of Tarka Samgraha. Presbyterian Mission Press, Allahabad (1849), http://books.google.com
6. Ganeri, J.: Indian Logic: A Reader. Routledge (2001)
7. Mahalakshmi, G.S., Anupama, N., Chitra, R., Geetha, T.V.: Deepika – A Non-Monotonic Reasoning System Based On Indian Logic. In: International Conference on Soft Computing techniques in engineering, SOFTECH – 2007, Avinashilingam University for Women, Coimbatore, India, pp. 470–476 (January 2007)
8. Mahalakshmi, G.S., Geetha, T.V.: Reasoning and Evolution of consistent ontologies using NORM. In: IJAI, Indian Society for Development and Environment Research (ISDER), vol. 2(S09), pp. 77–94 (Spring, 2009); ISSN 0974-0635
9. Vidyabhusana, S.C.: A History of Indian Logic – Ancient, Medieaeval and Modern Schools, p. 84. Motilal Banarsidass Publishers Private Ltd., Delhi (1988); ISBN:81-208-0565-8
10. Swami Virupakshananda, Tarka Samgraha, Sri Ramakrishna Math, Madras (1994)
11. Wada, T.: Invariable Concomitance in Navya-Nyaya, Sri Garib Dass Oriental Series No. 101. Indological and Oriental Publishers, New Delhi (1990)
12. Vanderwende, L., Kacmarcik, G., Suzuki, H., Menezes, A.: MindNet: An Automatically-Created Lexical Resource. In: Proceedings of HLT/EMNLP 2005 Interactive Demonstrations, Vancouver, British Columbia, Canada (October 2005)
13. Concept Net, `http://www.conceptnet.org`

Formal Definitions of Reason Fallacies to Aid Defect Exploration in Argument Gaming

G.S. Mahalakshmi and T.V. Geetha

Department of Computer Science and Engineering,
Anna University, Chennai 600 025, Tamil Nadu, India
{mahalakshmi,tvgeedir}@cs.annauniv.edu

Abstract. Reason fallacies are fallacious reasons presented in arguments during argumentative discussions. The fallacious reasons do not support the claim of argument and therefore, the argument gets defeated. Defect exploration is the process of analysing argument inconsistencies due to the presence of fallacious reasons. The context of argument exchange is a knowledge base represented in the form of Indian logic based ontology, and defect exploration actually entails analyzing the ontological elements of submitted arguments, during the 'tarka' style of argumentation. The process of defect exploration exploits the Navya-Nyaya methodology for identifying defects by exploring the presence or absence of invariable concomitance relation between elements of submitted arguments and populates them into a defect set. The defect set can then be utilized by the participant for designing appropriate defeat strategies, which help in generation of associated counter-arguments. In this paper, we propose the formal definitions of reason fallacies in terms of elements of arguments recommended by Nyaya logics, so that, the arguments are thoroughly analysed in a 'tarka' based argumentation setting.

Keywords: Indian logic, Nyaya Sastra, Reason fallacies, Defect, Argumentation.

1 Introduction

The basis of 'tarka' based argumentation for knowledge sharing, also referred to as argument gaming, is the use of game theoretic framework for rational exchange of arguments between two participating knowledge-sharing entities. Inferring valid information through argumentation procedures and assigning rewards for such inferences plays important role in allowing game theory to fit into the argument gaming scenario. Fallacies are logical errors, which are to be avoided in any discussion. A fallacy is an object of knowledge that obstructs an inference [13]. Presence of fallacies in an argument makes the argument defective [1]. Overcoming these fallacies or defects is called "removing the holes" from the submitted argument [14]. A fallacy is a component of an argument [4] which, being demonstrably flawed in its logic or form renders the argument invalid in whole.

The syllogistic method of argument formation as cited in Nyaya philosophy [14] states the possibility of fallacies or defects that are found identifiable in stated arguments,

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 243–256, 2009.

through which any argument can be challenged for justification and existence. (Fallacies can also be present with the manner in which the argument is proposed. Such fallacies are called Argument Fallacies [7], which is not our scope of discussion).

Indian philosophy proposes various fallacies which inhibit the existence or proof of an argument. Those fallacies are called reason fallacies [14]. Reason fallacies originate from the reason (or) probans of the submitted argument [13]. Reason is something which is stated to support or prove the existence of object to be inferred over the context subject. Fallacies present in the reasons may dilute or weaken the argument, or, in other words, they may fail in proving the object of inference. Therefore, reason and its relation with other elements of argument are of prime importance in argumentative discussions.

Indian philosophy classifies the structure of argument into three parts: Subject, the object to be inferred or probandum, and the reason or probans [3]. During scholarly debates, the opponent always attempts to find any flaws related to the argument focusing the probans element [5]. We refer to the flaws as defects or holes of the argument [8]. These defects may be well within the statement of probans or the relation of probans with other elements of arguments. The process of attempting to identify the nature and type of defects is called defect exploration. [8]. In the attempt to identify the defect present in the argument, there may be more than one defect detected which is populated into the defect set of that particular argument.

Any defect identified with the submitted argument qualifies the argument for attack or refutation. Refutation can be defined as pointing out the defects or fallacies in the statements of the opponent, which causes defeat to the argumentator [6]. There may be more than one refutation due to the presence of more number of defects in the submitted arguments. The best refutation [10] qualifies to be a counter-argument. This best describes the argument gaming scenario.

2 Related Work

Fallacies are something that weakens the argument. To make the argument more logical or stronger, these fallacies should be avoided while constructing the argument. Fallacies can be formal or informal. P. Ikuenobe [12], in his work on nature of fallacies argues like this:

> *An argument involves the formal and informal processes or methods of proving or establishing for ourselves or others that a proposition or belief as the conclusion of an argument has adequate and relevant evidential basis for its acceptance. The vague epistemic foundation of adequacy in justification, as a plausible basis for unifying fallacies, implies that there are different degrees, varieties, and guises of the epistemic failure of proof [12, page 194].*

Formal fallacies deal with the manner in which the argument is proposed or the structure of the argument; and informal fallacies deal with the content of the argument. There are different types of formal and informal fallacies. To identify the type of the fallacy, the nature of fallacies should be clearly understood. Several literatures exist in the arena which makes critical discussion about the understanding of argument fallacies as formal or informal [11]. A large part of such study of fallacies can be seen in the works of Walton [16] and Woods and Walton [17].

A different approach to understanding and classifying fallacies is provided by argumentation theory [2]. In this approach, an argument is regarded as an interactive protocol between individuals which attempts to resolve a disagreement. The protocol is regulated by certain rules of interaction, and violations of these rules are fallacies.

Walton [16] defines a fallacy as follows:

> *A fallacy is (1) an argument (or at least something that purports to be an argument); (2) that falls short of some standard of correctness; (3) as used in a context of dialogue; (4) but that, for various reasons, has a semblance of correctness about it in context; and (5) poses a serious obstacle to the realization of the goal of a dialogue (p. 255).*

For Walton, a fallacy is fundamentally negative; it involves a lapse, error, failure, and deception. According to Walton [16], it is exactly how such failures [in fulfilling a burden of proof] occurs, by what means, that determines which fallacy occurred or whether a fallacy occurred (16, p. 10). Some of the elements of a fallacy are that it involves the use of a systematic deceptive technique in the context of a dialogue. The use prevents one from achieving the desired goals of various forms of dialogue: the goal depends on whether the dialogue is a negotiation, persuasion, critical discussion, deliberation, or inquiry.

All the above literatures on argumentation dialogues involved some or the other way of identification of fallacies present in the arguments. However, Indian philosophy takes a different perspective of approaching the definition of fallacies present in the arguments. i.e. while all the above literatures examined the argument fallacies (both formal and informal), Indian philosophy attempted to explore more on the probans because probans or reason is the most supportive part of proof of the argument. Hence, fallacies of arguments in Indian philosophy are termed as 'reason fallacies' [3, 14]. Attempting to interpret an argument with a defective reason will result in fallacious reasoning of that argument. The following section describes the utilization of defects in argument gaming.

3 Defects in Argument Gaming

Indian philosophy defines several types and sub-types of reason fallacies [14] which can be utilized in the process of defect exploration [8] in argument gaming. Defects or reason fallacies, when identified from an argument, and when exploited to generate the next immediate counter-argument, will contribute greatly in interpretation of the submitted argument. Therefore, generation of defects is the driving force behind reasoning from arguments. Generation of defects could be appropriate if and only if the submitted argument is interpreted in the right sense. In argument gaming, we interpret arguments using 'Nyaya logics'. Nyaya logics are a mechanism for defining arguments according to Indian philosophical view point [9]. According to Nyaya logics, every argument is realized in terms of subject, object of inference and reason with internal relations between them. Arguments, when analysed with Nyaya logics, reveal the presence of defects in the elements of arguments. The constituent elements of arguments are compared with the knowledge base of the arguing authority to reveal the exact number and nature of defects, as connected to the enriched concept and relation elements of the Indian logic based ontology. Evaluation of defect observations result in

assignment of argument rewards. To maximize rewards, effective inferences have to be performed at every argument exchange, which demand complete observations of defects from the submitted argument.

The prime factor beside argument defect exploration is the nature of a special relation, invariable concomitance [15]. The defect exploration algorithm exploits the Navya-Nyaya methodology for identifying defects by exploring the presence or absence of invariable concomitance relation between elements of submitted arguments and populates them into a defect set. The defect set can then be utilized by the participant for designing appropriate defeat strategies, which help in generation of associated counter-arguments. As we have already described (refer Section - Introduction) continuous repeated argument process of defect exploration, defeat strategy determination and counter-argument generation in a game theoretic framework will eventually result in knowledge sharing between the participating entities. The core concept behind defect analysis is invariable concomitance, which shall be described in the context of Nyaya based ontology. In the following section, we record invariable concomitance from the viewpoint of defect analysis and go on to show how invariable concomitance contributes to argument gaming model for knowledge sharing.

4 Invariable Concomitance – The Theory Behind Defect Analysis

Invariable concomitance relation is one of the shared conceptualization of Nyaya, which aids primarily in the proof of object of inference over the given subject [15]. Generally, relations exist between a concept pair and/or between a concept and its member qualities. The types of relations play a major role in determining the type of defects. Presence of invariable nature of relation between concepts overrides any other relation that is said to exist between them. Various deviations such as the absence of a relation where it needs to exist, presence of prohibited concept/relation elements, absence of necessary concept/relation elements, conditional presence or absence of necessary elements etc. contribute to projecting holes or defects out of the submitted argument.

The identified holes are categorized into defect categories based primarily on whether the element is basically a concept or a relation. Other sub-categorical information under concept/relation category like direct/indirect presence of relations, invariable nature of relation etc. are also used. In knowledge sharing by procedural arguments, the presence or absence of invariable concomitance relation and its type are recorded implicitly during argument analysis. The defects present in the arguments are accumulated into a hole set or defect set. By the above approach, indirect inference of world knowledge embedded in the argument is captured and stored in the form of conceptual ontology. The following section presents a detailed overview of defects as per Nyaya Sastra.

5 Defects

5.1 Overview of Defects

An argument, according to Nyaya, consists of two major components, viz., concept and relation. Concept can be further sub-divided as subject, reason or object of inference.

Relation can be further sub-divided into relation between subject and reason, relation between reason and object of inference, and relation between subject and object of inference. According to Nyaya, defective reason is classified into five types, depending on which component of the argument contributes to the defect. They are viz. (1) the erratic or uncertain or straying (2) the contradictory or adverse (3) the counterbalanced or antithetical (4) the unproved or inconclusive or unestablished (5) the incompatible or stultified [13, p. 38]. The defects involved in the above reasons are respectively the following: (1) the erraticalness or uncertainty (2) contradiction (3) counterbalance (4) absence of proof or inclusiveness, and (5) incompatibility.

5.1.1 The Erratic Reason

The erratic or Straying, is a reason or middle term in which abides a character, the possession of which causes that presence of two alternatives which produces doubt in the probandum or the major term. This defective reason is further subdivided into (1) that which is too general, referred to as Straying:common, (2) that which is non-general or not general enough, referred to as Straying: uncommon, and (3) that which is non-exclusive, also known as non-conclusive.

5.1.2 The Contradictory Reason

The contradictory or adverse is a reason which is the counter-part of that non-existence which constantly accompanies the major term. In other words, it can be defined as a reason which is constantly accompanied by the absence of the probandum, the major term.

5.1.3 The Counter-Balanced Reason

If, at the time of consideration of a reason which seeks to establish the existence of the probandum or major term, there occurs the consideration of another reason which seeks to establish the non-existence of that term, then, both the reasons are said to be counter-balancing each other. i.e. the inference from one reason being of as much force as that from the other reason, the two inference neutralise each other. The counter-balanced reason is also known as antithetical reason.

5.1.4 The Unproved Reason

The unproved reason is also known as unestablished reason. It is of three kinds: (1) unproved on the part of its locus or the subject, also known as unestablished to subject, (2) unproved with regard to its own nature, or unestablished to itself, and (3) unproved in respect of accompaniment, or unestablished to concomitance.

5.1.5 The Incompatible Reason

An incompatible reason or stultified reason occurs when there is the knowledge that the major term, which is assigned to the minor term, does not really abide in it.

5.2 Classification of Defects

Fallacies are serviceable as they point out inefficiency. A fallacy when exposed is a good reply to an opponent, whose argument is thus pointed out to be inefficient. However, in order to analyse defects in terms of presence and absence of elements of

arguments, we propose here, a two-dimensional classification table, called as defect table, adapted based on the above five defect categories, defined traditionally in Nyaya Sastra. To build the defect table, the classification of defects is proposed, which is based on the component of argument that contributes to that particular defect.

Concept and relation are the two major broader classes of elements of arguments. Therefore, we tend to classify the above-mentioned five defects and their internal categories as, concept-based defects, and relation-based defects (refer Fig. 1).

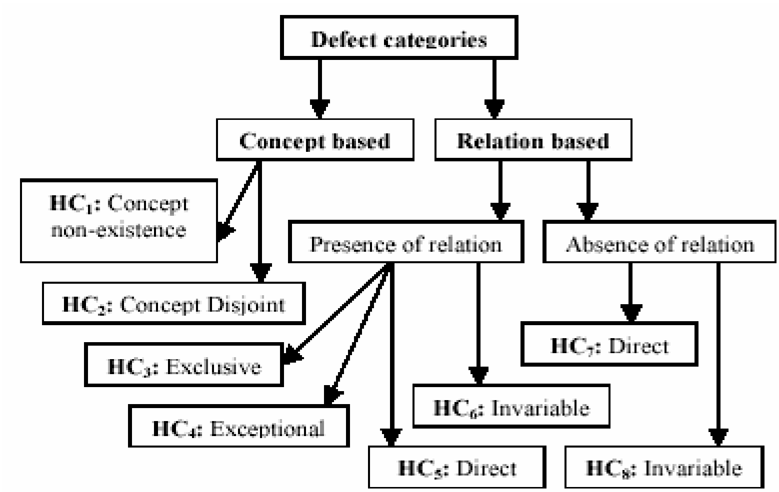


Fig. 1. Categorisation of Defects

The non-existence of concept is tabulated under concept-based defects. Non-existence is assumed to be absolute non-existence [V. Swaminathan, 2001]. The presence or absence of relations are analysed under relation based defect categories. The relation present may be of type exclusive, exceptional, direct or invariable; Absence of relation may be direct absence or invariable absence. Table 1 illustrates the nature of concepts and relations in their defective form, as they occur in any argument. The operator '!' indicates non-existence and '~' indicates negation supported by a proof. The prefixes I, D, X, Xp stands for Invariableness, Direct, Exclusive and Exceptional type of relations.

The defect table in Table 1. maps the traditional defect types with our proposed defect categories in Fig. 1. In this context, we go on to explain the interpretation of defect table. As discussed earlier (refer introduction), a clear analysis of argument concepts and their relations will generate a defect summary for every argument at hand. By adapting the traditional definition of defect types, the entries in the defect table can now be interpreted as follows. If the relation existing between *concept: reason* and the *non-existing concept: object of inference* is invariable, then there exists a defect *Adverse* with the relation element of argument; if the relation is of type direct, then it is said to fall under *Straying-Common*. Presence of exclusive or exceptional type of relation from *concept: reason* with *concept: subject* is said to count for defect, *Straying-Uncommon*.

Table 1. Defects per Hole Category in terms of Elements of Arguments

Reason	Straying					Unestablished			
Hole category	**Common**	**Uncommon**	**Non-conclusive**	**Adverse**	**Antithetical**	**To subject**	**To itself**	**To invariable**	**Stultified**
HC_1	$!C_{OI}$			$!C_{OI}$		$!C_S$			$\sim C_{OI}$
HC_2					C_{OI}, $!C_{OI}$				C_{OI}, $\sim C_{OI}$
HC_3		$X\text{-}R_{R\text{-}S}$							
HC_4		$Xp\text{-}R_{R\text{-}S}$							
HC_5	$R_{R\text{-}!OI}$		$R_{R\text{-}\{S\}}$		$R_{R1\text{-}OI}$ $R_{R2\text{-}!OI}$				$R_{R1\text{-}OI}$
HC_6				$I\text{-}R_{R\text{-}!OI}$					
HC_7							$!R_{R\text{-}S}$		
HC_8								$!I\text{-}R_{R\text{-}OI}$	

If a direct relation is present from *concept: reason* with every *concept: subject* of the ontology leaving no subject behind for comparison, then the defect *Straying: non-conclusive* is said to exist. When both the existence and non-existence of the *concept: object of inference* is allowed in the ontology, with one *concept: reason* supporting for the existence and another equally qualifying *concept: reason* supporting for the non-existence, then the defect *Antithetical* is said to exist with such type of reasons. A little variant of this is the defect Stultified. For this defect to exist, the existence and negation of *concept: object of inference*, both are allowed, but the negation of *concept: object of inference* should be supported by a proof with respect to the ontology. In such a situation, the presence of direct relation between the *concept: reason* with that of the *concept: object of inference* is said to qualify for defect *Stultified*. Absence of invariable relation between the *concept: reason* with the *concept: object of inference*, is said to fall under the hole category, *Unestablished to invariable*.

Above all, there should exist a relation existing from the *concept: reason* to the *concept: subject* upon which the existence of *concept: object of inference* needs to be analysed, which is the fundamental notion of defect exploration. If that relation between the *concept: reason* with the *concept: subject* does not exist, then there is no room for proving the existence of object over the subject. In other words, the middle term, the *concept: reason* has no connection with the *concept: subject* and hence, the defect *Unestablished to itself* is said to occur in the argument. All the above defects referred deal with only the relation element of *concept: reason* with other concepts of

the argument. There is another defect, which is only concept-based. That is, the absence of *concept: subject* upon which the entire defect exploration is carried over from every other perspective generates a defect *Unestablished to itself* because such a *concept: subject* never exists. The following section provides a more formal definition of defect and defect types.

5.3 Formal Definition of Defects

Definition 1 (Defect). According to Nyaya school of Indian logic, a fallacy is an object of knowledge which obstructs an inference, and is known as defective reason [13]. In general, these defects are a simple combination of concept and/or relation elements of the argument. From this perspective, we have categorised the defects into two major divisions: concept-originating, relation-originating. The relation-originating defects may be arising either due to the presence of recommended relation at a wrong place or absence of a required relation at places where it is required. The defect exploration algorithm looks for existence of concepts and the nature of relations between concepts to identify the class and type of defects existing in the submitted arguments. In addition, the attributes of concepts and relations are also analyzed for occurrence of defects.

Defective reasons are basically of five types [13]. They are adverse, straying, antithetical, unestablished and stultified. Each of these defects are defined as follows:

Definition 2 (Adverse Defect). An input argument A is said to have Adverse defect iff $c_1 \vDash c_2$ for every $c_1 \in C_{R,}\ c_2 \in C_{OI}$, $C_{OI} \not\subset \mathcal{E}$.

If the relation existing between *concept: reason* and the *non-existing concept: object of inference* is invariable, then there exists a defect *Adverse* with the relation element of argument. (Note: $\mathcal{E}$ is the knowledge base)

Definition 3 (Straying:Common Defect). An input argument A is said to have Straying:Common defect iff $c_1\ r\ c_2$ for every $c_1 \in C_{R,}\ c_2 \in C_{OI}$, $C_{OI} \not\subset \mathcal{E}$.

If the relation existing between *concept: reason* and the *non-existing concept: object of inference* is direct, then it is said to fall under *Straying-Common*.

Definition 4 (Straying:UnCommon Defect). An input argument A is said to have Straying:UnCommon defect iff $c_1\ r\ c_2$ for every $c_1 \in C_{R,}\ c_2 \in C_S$, $r \in \{\prec, \preccurlyeq\}$.

Presence of exclusive or exceptional type of relation from *concept: reason* with *concept: subject* is said to count for defect, *Straying-Uncommon*.

Definition 5 (Straying:Non conclusive Defect). An input argument A is said to have Straying:Non conclusive defect iff $c_1\ r\ c_2$ for every $c_1 \in C_{R,}\ c_2 \in \{\{C_S\} \subseteq O_T\}$.

If a direct relation is present from *concept: reason* with every *concept: subject* of the ontology leaving no subject behind for comparison, then the defect *Straying: nonconclusive* is said to exist. (Note: O_T is the total concepts in the ontology which represents the knowledge base).

Definition 6 (Antithetical Defect). An input argument A is said to have Antithetical defect iff $c_1\ r\ c_2$ and $c_3\ r \neg c_2$ for every $c_1, c_3 \in C_{R,}\ c_2 \in C_{OI}$.

When one *concept: reason* supporting for the existence and another equally qualifying *concept: reason* supporting for the non-existence, then the defect *Antithetical* is said to exist with such type of reasons.

A little variant of this is the defect Stultified. For this defect to exist, the existence and negation of *concept: object of inference*, both are allowed, but there is the presence of direct relation between the *concept: reason* with that of the *concept: object of inference* is said to qualify for defect *Stultified*.

Definition 7 (Stultified Defect). An input argument A is said to have Stultified defect iff $c_1\ r\ c_2$ for every $c_1 \in C_R$, $c_2 \in C_{OI}$, $\{ C_{OI}, \sim C_{OI} \} \subseteq \mathcal{E}$.

Definition 8 (Unestablished to Invariable Defect). An input argument A is said to have the defect unestablished to invariable iff c1 ⊭ c2 for every c1 ∈ CR, c2 ∈ COI. (Note: ⊭ should be read as 'not invariably related to').

Absence of invariable relation between the *concept: reason* with the *concept: object of inference*, is said to fall under the hole category, *Unestablished to invariable*.

Above all, there should exist a relation existing from the *concept: reason* to the *concept: subject* upon which the existence of *concept: object of inference* needs to be analysed, which is the fundamental notion of defect exploration. If that relation between the *concept: reason* with the *concept: subject* does not exist, then there is no room for proving the existence of object over the subject. In other words, the middle term, the *concept: reason* has no connection with the *concept: subject* and hence, the defect *Unestablished to itself* is said to occur in the argument.

Definition 9 (Unestablished to itself Defect). An input argument A is said to have the defect unestablished to itself iff $c_1 \nvdash c_2$ for every $c_1 \in C_R$, $c_2 \in C_S$. (Note: $\nvdash$ should be read as 'not related to').

All the above defects referred dealt with only the relation element of *concept: reason* with other concepts of the argument. There is another defect, which is normally concept-based. That is, the absence of *concept: subject* upon which the entire defect exploration is carried over from every other perspective generates a defect *Unestablished to subject* because such a *concept: subject* never exists. Alternatively, the *concept: subject* which is referred to in the argument, may be present with the knowledge base but, the relation of *concept: subject* to *concept: reason* referred in the argument may not actually exist or may be negated for that *concept: subject.* In such cases too, it can be said that the submitted argument has the defect, *Unestablished to subject* with respect to the knowledge base.

Definition 10 (Unestablished to subject Defect). An input argument A is said to have the defect unestablished to subject iff $c_1\ r\ c_2$ for every $c_1 \in C_R$, $c_2 \notin \{\{C_S\} \subseteq O_T\}$.

The definition of defects (which maps both traditional defect classification and our proposed defect categorization) is formally supplied in the form of defect table. The defect exploration algorithm utilizes the defect table to decide on the class and category of the identified defect from the submitted argument. The following section unveils the technique of defect exploration from procedural arguments in a more detailed manner.

6 Defect Exploration

The process of exploration of defects has three phases: Defect Exploration, Defect classification and Defect Identification. Defect exploration [8] consists of initially analyzing a given input argument and later highlighting the argument's defects or holes in terms of elements of arguments. As the defects are explored, the class and category to which every defect belongs to, is identified and tabulated. The algorithm initially splits the submitted argument in terms of its constituent elements of arguments. After argument analysis the defect exploration algorithm looks for existence of concepts and the nature of relations between concepts to identify the class and type of defects out of the submitted arguments. The quality of relation is also analyzed for occurrence of defects. The elements of input argument are tabulated (by referring to Table 1) across various defect categories. From this detailed representation, every argument is analyzed for presence of defects.

The algorithm primarily looks for the existence of concepts, reason and subject, in order to identify concept-related defects. Otherwise, the input argument is said to undergo a thorough screening for the checking of relations between the concepts present and the type and nature of relations that exist between them. Every defect capture lists the defect type(s) and defect name within the element of argument. After defect exploration, the results are summarized into a hole set or defect set.

Let us assume, the arguer proposes an argument which is expected to contain, generally, the subject, reason and the object to be inferred along with the relations existing between them. Presence of reason is a mandatory thing for an argument to be considered for discussion, because, the reason is said to provide a support for proving the existence of object of inference over the referred subject. If the input argument has no reason component, then, there is no purpose in arguing for or against that argument, because the argument itself is incomplete.

If the knowledgebase of the counter-arguer does not contain the subject component listed in the argument of the arguer or proposer, then there is the defect *unestablished to subject*. If the reason and subject components of arguments have no relation with each other according to the counter-arguer's perspective, then there exists the defect, *unestablished to itself* within the proposed argument. In a similar case, if the reason and subject component of the proposed argument are actually exclusively or exceptionally related to each other, and the proposer has not utilized those or violated the definitions of exclusive or exceptional relations while proposing the argument, then, we say, the argument is very uncertain with a special case. The defect is known as *straying-uncommon*.

If the reason is related to more than one subject, and, if those subjects to which the reason relates to, are members of a set, which share atleast a commonality between one another, then, the existence of object of inference over a particular subject shall not be realized because, the proposed reason argues the presence of object of inference, not to a subject, but to a class of subjects. The proposed argument is still uncertain and the defect is referred as *straying-nonconclusive*.

If the object to be inferred stated in the proposed argument has a reversible presence in the knowledge base of the counter-arguer, i.e., if the argument proposes the existence of object of inference over the subject, by supporting it with a reason, and, if that object of inference is already proved to be non-existing in the knowledge base of the counter-arguer, then, we say, the defect *stultified* is present in the proposed

argument. Because, at this point, the counter-arguer is unable to decide which state of existence (presence or absence) of object of inference is valid.

If the argument supplies a reason (reason1) for the existence of object of inference, and the counter-arguer has another reason (reason2) recorded in its own knowledge base which supports the non-existence of object of inference over the subject, and, if both the reasons are directly related to the respective state of object of inferences, then, we say, the defect *antithetical* is said to exist. In other words, if there are two different reasons, one which directly supports the existence of object of inference as mentioned in the submitted argument, and the other, which directly supports the non-existence of object of inference according to the counter-arguer's opinion, then, both the reasons said to counter-balance each other. Therefore, this defect is known as counter-balanced or *antithetical*.

If the argument states the existence of object of inference by supporting with a reason, and in the counter-arguer's perspective, that reason and object of inference have no direct relation with each other, then the argument loses its strength. The support provided by the arguer is a weak support. The argument has to be rejected blindly from discussion. If the reason stated in the argument as a support for object of inference, is invariably related to the object of inference, then, the proposed argument shall be accepted beyond a formal proof, because of the invariable presence of reason with object of inference. As discussed earlier, in section 1, the presence of invariableness is a special mechanism utilized for proving arguments in discussions.

If the arguer proposes only a single reason for the support of object of inference, thinking that the invariableness between both would prove the entire argument and if there is no invariable connection between the reason and object of inference, according to the knowledge perspective of the counter-arguer, then, here comes a difference of opinion with respect to the invariable relation between arguer and counter-arguer. Both agree that they knew about every other information related to the proposed argument but does not agree with respect to invariable concomitance. Therefore, from the viewpoint of counter-arguer, the argument is said to have the defect, *unestablised to invariable concomitance*.

If the argument supports the existence of object of inference with a valid reason, and if the same reason is supporting only the non-existence of the same object of inference, according to the counter-arguer's knowledge base, then, there is a conflict between the arguer and counter-arguer. The situation is very uncertain, as to whose knowledge is valid. This is a very common situation where we see people objecting to other's opinions. Therefore, the counter-arguer may refer to the defect as *straying-common*.

If the argument supports the existence of object of inference with a valid reason, and if the same reason is invariably related to the non-existence of object of inference, according to counter-arguer, then, both of them are said to contradict each other's opinion. We refer to this defect as *adverse*.

The following section provides a case study for revealing defects from sample arguments used in argument gaming.

7 Case Study

We formulated 9 sample arguments over which the defect analysis is performed and the results are tabulated (refer Tables 2 and 3). These arguments are not specialised in

any domain and carry simple statements which revolve around world knowledge of any arguing entity. If the information is not found in the knowledgebase the maximum weight of a concept / relation in the knowledge base is given as a defect value. From implementation statistics, we shall conclude that, defect value is max. when the knowledge base is refreshed (and a defect is found) on a larger scale.

Table 2. Splitting of Argument Elements in sample arguments

Arg. Id	**Argument**	**Subject**	**object of inference**	**reason**
1	sky_lotus has fragrance	sky_lotus	fragrance	Nil
2	artificial-rose has fragrance	artificial-rose	fragrance	Nil
3	lily has fragrance	lily	fragrance	Nil
4	mountain has fire due_to smoke	mountain	fire	smoke
5	penguin fly because it is-a bird	penguin	fly	Bird
6	bats are viviparous because they are mammal	bat	viviparous	mammal
7	Falls does not have fire when there is smoke	falls	fire	smoke
8	Falls does not have fire when there is smoke	falls	fire	smoke

Table 3. Argument Defects

Arg. Id	**Status in KB**	**Defect Category & Type**	**Status in KB**
1	concept doesn't exists	HC1 Unestablished to subject	concept doesn't exists
2	concept exists, fragrance as a quality(negation)	HC7 Unestablished to itself	concept exists, fragrance as a quality(negation)
3	concept and quality exists	No Defect	concept and quality exists
4	Fire, smoke exists as concepts. No invariable relation	HC8 Unestablished to invariance	Fire, smoke exists as concepts. No invariable
5	Penguin and bird exists as concept. Exclusive quality: fly in negation	HC4 Straying Uncommon	Penguin and bird exists as concept. Exclusive quality: fly in negation
6	Bat, mammal and bird exist as concept. Mammal-viviparous, bird-~viviparous	HC2, HC5 Antithetical	Bat, mammal and bird exist as concept. Mammal-viviparous, bird-~viviparous
7	Falls and smoke exist as concept. Absence of fire as concept. Direct relation between fire and smoke	HC1, HC5 Straying Common	Falls and smoke exist as concept. Absence of fire as concept. Direct relation between fire and smoke
8	Falls and smoke exist as concept. Absence of fire as concept. Invariable relation between fire and smoke	HC1, HC6 Adverse	Falls and smoke exist as concept. Absence of fire as concept. Invariable relation between fire and smoke

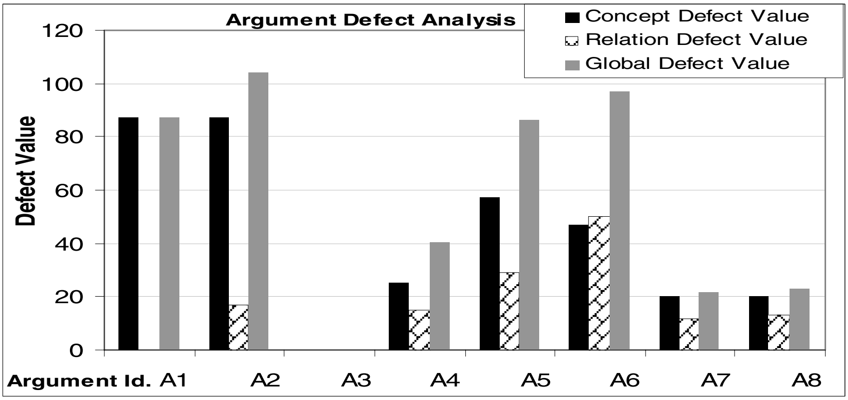


Fig. 2. Defect Analysis - Results

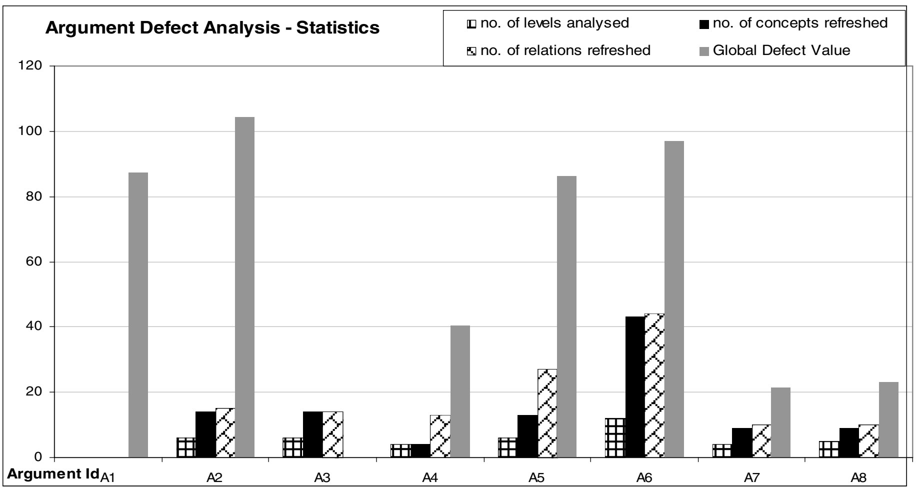


Fig. 3. Ontology Statistics of Argument Defect Analysis

Therefore, besides identifying the defects [8], the defect exploration algorithm [8] also classifies the identified defects into various defect types according to the element of argument from which the defect has originated. Figure 3 depicts the quantity of concepts and relations refreshed during argument analysis and Figure 2 depicts the harvested defect values. The levels of knowledge refreshed in the knowledge base also contribute to the analysis of defects in arguments.

8 Conclusion

The aim of this paper is to obtain a formal definition for reason fallacies present in the arguments. The formal definitions were inspired by Indian logic approach of interpreting arguments during 'tarka' methodology of argumentative discussions.

The objective is to utilize the presence of fallacies in the reason or probans of the submitted argument for further generation of counter-arguments. In future, introducing argument fallacies from the western philosophy, identifying the conceptual connection of western and Indian methodology of fallacies of argumentation and arriving at a hybrid argument gaming model is of our interest.

References

1. Chakraborti, C.: Logic: Informal, Symbolic and Inductive. Prentice Hall of India, New Delhi (2006)
2. Eemeren, F.H., Grootendorst, R.: Argumentation, communication, and fallacies. A pragma-dialectical perspective. Lawrence Erlbaum, Hillsdale (1992)
3. Vidyabhusana, S.C.: Gautama, The Nyaya Sutras. In: Sinha, N.L. (ed.) Sacred Book of the Hindus, Allahabad (1930); Reprinted in Motilal Banarsidass, Delhi (1990)
4. Hamblin, C.L.: Fallacies. Methuen, London (1970)
5. Hintikka, J.: Socratic Epistemology: Explorations of Knowledge-Seeking by Questioning, p. 239. Cambridge University Press, Cambridge (2007)
6. Ballantyne, J.R.: Lectures on Nyaya Philosophy-Embracing the text of Tarka Samgraha. Presbyterian Mission Press, Allahabad (1849), http://books.google.com
7. Mahalakshmi, G.S., Geetha, T.V.: A Mathematical Model for Argument Procedures based on Indian Philosophy. In: Proc. of International Conf. Artificial Intelligence and Applications (AIA 2006) as part of the 24th IASTED International Multi-conference Applied Informatics (AI 2006), Austria (2006)
8. Mahalakshmi, G.S., Geetha, T.V.: Navya-Nyaya Approach to Defect Exploration in Argument Gaming for Knowledge Sharing. In: Proc. of International Conf. on Logic, Navya-Nyaya & Applications - A Homage To Bimal Krishna Matilal (ICLNNA 2007), Jadavpur Univ., Calcutta (2007)
9. Mahalakshmi, G.S., Geetha, T.V.: Reasoning and Evolution of consistent ontologies using NORM. International Journal of Artificial Intelligence (IJAI), Indian Society for Development and Environment Research (ISDER), India (2008a)
10. Mahalakshmi, G.S., Geetha, T.V.: Modeling Uncertainty in Refutation Selection – A POMDP based approach. Journal of Uncertain Systems - special issue on Advances in uncertain theory and its applications (2008)
11. Powers, Lawrence: The One FallacyTheory. Informal Logic 17(2), 303–314 (1995)
12. Ikuenobe, P.: On the Theoretical Unification and Nature of Fallacies, Argumentation, vol. 18, pp. 189–211. Kluwer Academic Publishers, Netherlands (2004)
13. Vidyabhusana, S.C.: A History of Indian Logic – Ancient, Medieaeval and Modern Schools, p. 84. Motilal Banarsidass Publishers Private Ltd., Delhi (1988)
14. Swami Virupakshananda: Tarka Samgraha, Sri Ramakrishna Math, Madras (1994)
15. Wada, T.: Invariable Concomitance in Navya-Nyaya, Sri Garib Dass Oriental Series No. 101. Indological and Oriental Publishers, New Delhi (1990)
16. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning, pp. 165–167. SUNY Press (1995)
17. Walton, D., Woods, J.: Argument: The Logic of the fallacies, ch. 2. McGraw-Hill Ryerson, Toronto (1982)

The Art of Non-asserting: Dialogue with Nāgārjuna

Marie-Hélène Gorisse

Department of Philosophy
University of Lille 3
59000 Lille
France
mhgorisse@gmail.com

Abstract. In his excellent paper, *Nāgārjuna as anti-realist*, Siderits showed that it makes sense to perform a connection between the position of the Buddhist Nāgārjuna and contemporary anti realist theses such as Dummett's one. The point of this talk is to argue that this connection is an important one to perform for one's correct understanding of what Nāgārjuna is doing when he criticizes the contemporary Indian theories of knowledge and assertion, first section, but as soon as the theories of argumentation are involved, this connection can be implemented in a better way from an other anti realist perspective, namely the one of Dialogical logic (Erlangen school), in which the signification is given in terms of rules in a language game.

The philosophical issues are to hold an interpretation of the type of assertion performed by Nāgārjuna. We here aim at making a rational reconstruction of his chief claim 'I do not assert any proposition' in which a proposition is considered as the set of its strategies of justification.

As for the last section, the point will be to apply these analyses to Buddhist practice. We will in this section consider the conventional character of human activities as the fact that any speech act is performed within a dialogue under *ad-hoc* restrictions; and the question of one's progress in the soteriological path to liberation will be asked[1].

1 Nāgārjuna on Theories of Knowledge

1.1 The Dependent Origination: An All-Inclusive Version of Causation

Nāgārjuna, one of the most influential thinkers of Buddhism and the founder of the *mādhyamika* school, the school of the Way of the Middle, developed in the second century AC a criticism of the contemporary Indian theories of knowledge and assertion. The key-concept of these criticisms is the concept of 'DEPENDENT ORIGINATION' (*pratītya-samutpāda*) as taught in the sūtras of the

[1] I would like to keenly thanks all members of the referee, which have helped me a lot throughout this reconstruction of Nāgārjuna's thought.

R. Ramanujam and S. Sarukkai (Eds.): ICLA 2009, LNAI 5378, pp. 257–268, 2009.

prajñā-pāramitā, the 'Perfection of Wisdom'. The dependent origination is a technical Buddhist expression involving particular definitions of the notions of causality and contingency: there is not a chain but a *web* of causation such that the existence of a thing is contingent upon the existence of every other thing. For example, the existence of a tree is dependent upon the existence of a seed and upon the existence of wind, water, ground, and so on. In turn, the existence of the seed itself is also dependent upon the existence of the tree from which it comes and upon the existence of wind, water, ground, and so on. The list of the causes and conditions of existence of a thing can never be ended. Now, this tree is, in Nāgārjuna's perspective, nothing but the set of its conditions of existence. Thinking that there is an independent tree is thinking that there is a closed set of such conditions of existence, which is misleading. The Buddhists tell us that we have to think of reality as a generalized web of such dependencies and that the task of enunciating them is a never-ending task.

1.2 The Epistemic Level

The question therefore arises concerning everyday life practice: how is it that we do talk about the world and that we do have knowledge that governs our practice? The Buddhist answer amounts to saying that there is a decision from the knowing subject to carve out in the generalized web of interdependencies that she will call 'an object'. Therefore, she is always engaged within the choices she has made when she perceived. From this, her own conceptions are always engaged when she knows a fact of the world. In other words, the facts of the world and the knowledge I have of them can in no way be independent from each other. As Jay Garfield[2] puts it:

> To say that an object lacks essence, the Madhyamika philosopher will explain, is to say, as the Tibetans like to put it, that it does not exist "from its own side" [...] that its existence depends upon us as well.

Now, one of the great consequences of the fact that my knowledge depends on the context in which it has been gained is that there is no such context as the universal one, in which the proposition at stake could have been firmly established. In other words, every proposition can be questionable from a different perspective. This is the main observation that is pointed out in the chief work of Nāgārjuna, namely, the *Mūla-madhyamaka-kārika*, the 'Fundamental Stances of the Middle Way'. In this work, he shows for each universally alleged knowledge statement of an other Indian school of thought that it is questionable.

We immediately understand that the statement saying that '*every proposition can be disputable from an other perspective*' is itself disputable. This is essentially in order to avoid this type of criticisms that Nāgārjuna wrote the *Vigraha-vyāvartanī*, the 'Treatise to Prevent from Vain Discussions'. But in these lines, he supplies with an answer to these criticisms far much interesting than the classical problem of self reference. We here aim at a rational reconstruction of the strategy of Nāgārjuna in this work.

---

[2] In [4, p. 220].

2 Nāgārjuna on Theories of Assertion

In the *Vigraha-vyāvartanī*, Nāgārjuna does not implement his ideas directly with regard to the process of acquisition of knowledge, but to the process of assertion and negation of a given thesis within a debate. The link between the two could easily be reformulated in the following way: the justification of an assertion in a philosophical debate is but the demonstration of the fact that what is asserted is rationally guaranteed, that is to say that it is the subject of knowledge.

2.1 The Art of Making No Assertion

A New Approach to Knowledge Statements. First of all, let us examine again the dependency between my beliefs and the facts of the world. In order to have a better idea of what is at stake, let us focus on current theories that deal with similar conceptions, notably the anti realist position. Anti realist philosophers claim that, since there is no transcendent state of affairs, being epistemically guaranteed can not amount to being in adequacy with reality. The same important consequence has to be drawn from the Nāgārjunian conception, according to which I can not know something that exists independently of my knowledge of it. From this, Nāgārjuna is committed to the position that it is possible to give an account for the process of acquisition of knowledge that is different from the account in terms of adequacy with reality. And this precisely because we can never be sure of what is reality *per se*.

At a second stage, anti realist philosophers developed a new conception of knowledge according to which being epistemically guaranteed amounts to being *justifiable*. Here, the justification of knowledge is a conventional matter, it is a coherentist and not a foundationalist process and allows for a plurality of justified types of knowledge. In terms of assertion, this means that 'φ *is true*' means 'φ *is justifiable*'. And the semantic anti realist position is leading to the recognition of a plurality of ways in which an assertion can be said to be 'justifiable'.

Our claim is that Nāgārjuna speaks in terms of justification too. Notably because Nāgārjuna is in line with the Indian tradition of argumentation:

- First of all, the structure itself of the *Vigraha-vyāvartanī* is argumentative. More precisely, no position is put forward without its set of justifications. What is more, these justifications consist in the answer to all potential attacks of a conceivable opponent.
- Moreover, within the classical Indian tradition, something is admitted as knowledge *if and only if* it has been gained by means of a *pramāṇa*, a 'criterion for justified knowledge'. As argued, successfully in my view, by Siderits in [9] and by Waldo in [11], Nāgārjuna does not in the *Vigraha-vyāvartanī* call into question the *possibility* but the *uniqueness* of the *pramāṇa* account. Actually, Nāgārjuna is himself using *pramāṇa*. What is interesting for our subject is that this theory of *pramāṇa* has its roots within a theory of CONSENSUS. More precisely, the Naiyāyikas, who are the main interlocutors of Nāgārjuna in the *Vigraha-vyāvartanī*, consider that the right process to discriminate between beliefs that are knowledge and beliefs that are not is a

consensus, whose task is to find an equilibrium between the beliefs one has about the world and successful practice. And, as I said, Nāgārjuna is happy with it, he is not denying this approach but he is indicating that it is in nature coherentist, on-coming from the decision of a scientific community, and not foundational, on-coming from the structure of the world itself.

Therefore, the kind of semantic anti realism we are advocating here is more in line with Brandom's INFERENTIALISM, as outlined in [2], than in line with Dummett's approach, advocated in [3]. The main difference between the two is that Brandom performs what could be called a 'social turn'. According to Brandom, the nature of assertion consists in the fact that in asserting, the speaker achieves the following institutional effect: she undertakes the responsibility of justifying her assertion. Following the lines of Brandom[3], the important point here is therefore to analyse assertion as commitment. We claim that what Nāgārjuna is saying, though different in nature, is governed by the same rules that govern Brandom's inferentialism, namely:

- The fact that the signification of a proposition cannot be specified independently from the subject who enunciates this proposition ; and a switch from a referentialist semantics to a semantics in terms of conditions of assertability. This is a consequence of Nāgārjuna's position that nothing is independent.
- A conception of the act of assertion in which '*to assert* φ' means '*to commit oneself to give justifications for* φ'. Here, the notion of justification becomes basic. This reading follows from the fact that no position in the *Vigraha-vyāvartanī* is advocated without the set of all the strategies one could need to defend it. In other words, any position advocated in this work is present along with the set of all the strategies needed to make the point in a given discussion. Notice here that strategies are not the same depending on the identity of the opponent at stake[4].

The Dialogical Approach. In order to advocate this, we are going to make use of a formalism, namely Dialogical Logic, whose format is very likely to express Nāgārjuna's approach. This formalism is very straightforward, first because of its dialogical presentation ; and secondly for its anti realist motivations. Other conclusive connections will be stressed in the course of this section. The approach of dialogical logic developped by Rahman, as shown by [8], is a modified version of the CONSTRUCTIVIST approach of Lorenz and Lorenzen (Erlangen school), [7] enhanced with a PRAGMATIST orientation. It deals with the features of semantic anti realism just mentioned and measures the signification of a sentence by means of its conditions of assertability, that is to say by means of the set of all the possible strategies when discussing the proposition expressed by the sentence in question.

[3] In [2], Chapter 6: 'Objectivity and the Normative Fine structure of Rationality'.
[4] For example, *reductio ad absurdum* arguments are used in the *Mūla-madhyamaka-kārika* against an *Abhidharmika* opponent, while *petitio principii* arguments are the tool in the *Vigraha-vyāvartanī*, where they are adressed to a *Naiyāyika* opponent.

More precisely, what is at stake in this approach by means of a formal proof is to establish the validity of a sentence (which content is a proposition). Non-formally speaking, a formal proof is a game between two players, respectively called the Proponent and the Opponent, that ends when all the justifications of the sentence at issue are given or when no further move is allowed. The mark of the validity of a sentence is the presence of a winning strategy. There is a winning strategy when the Proponent wins the dialogue whatever the choices of the Opponent may be.

We can already see that a very important feature of this dialogical approach of logic is the asymmetry between the Proponent and the Opponent. Here, only the Proponent is performing genuine *assertions*. This is due to the '*formal restriction rule*' according to which only the Opponent can assert atomic formulas or, to put it in a different way, can assert elementary justifications. It is important to keep in mind here that atomic formulas are the parts of a formula that is not analysable through logical tools. We therefore are unable to prove them logically. Yet, asserting them by presupposing them would be but justifying a proposition within a particular case and, once more, we here deal with validity and the dialogues are *formal* dialogues. The Dialogical proposal is therefore to allow the use of such an elementary justification *if and only if* the Opponent has conceded it.

Now, as the Opponent's role is to defeat the Proponent's assertion, he will perform as few concessions as he can and will introduce the minimal set of atomic formulas. Testing the formal justification of a proposition within this type of dialogue is thus like convincing the most acute interlocutor. Hence, when the Proponent wins, the set of plays of the Opponent represent but the construction of the minimal set of presuppositions needed in order to prove the validity of the sentence (in order to assert that a given proposition holds in all situations); they do not represent the moves of a 'real' player. Let us remark here that the builder of the minimal set of conditions of assertability is itself by definition asserting without restriction at all. This is why he is therefore not to be considered as to be performing assertions.

What we will recall of this Dialogical approach for our purpose are the following features:

- A formal proof is put in the form of a dialogue, that is to say of a linguistic interaction. This is part of the pragmatist sensitivity.
- By means of the moves of the pseudo-player Opponent in a dialogical play, asserting a proposition amounts to asserting only the set of its justifications, that is to say, the set of its conditions of assertability. This is a consequence of the constructivist anti realist approach.
  From this, propositions are the forms of an achieved dialogue[5]. Hence, asserting a proposition amounts to asserting the entire dialogue that was used to assert it. As Keiff [6] put it:

[5] This conception is shared by linear logic and ludic logic as developed by Girard and associates (see for example [5]). The study of these frameworks could also be very fruitful to our understanding of Nāgārjuna's position.

The fundamental idea of Dialogical semantics could be enunciated through the following principle:
"The signification of an assertion which has its type entirely given by the form of the dialogue in which it has been asserted by a speaker, provides, for a critical interlocutor, with all the necessary justifications to succeed in his assertion"

Interpreting Nāgārjuna. These features of the Dialogical approach lead us to conceive it as a really adapted tool for one who wants to discuss Nāgārjuna's position on theories of assertion. Indeed, if a proposition is a dialogue brought to fruition, then any dialogue whose initial thesis is defective DOES NOT CONTAIN ANY PROPOSITION IN THAT VERY SENSE[6] because:

To assert is to commit oneself to provide with justifications

And in a formal game, this means to provide with justifications *in any situation*. In the Indian context, we deal with such a formal game when we speak about the process of the justification of an inference because an inference is a tool to get assured knowledge, that is to say unquestionable knowledge. This universality seems to be exactly what Nāgārjuna refutes when he says:

मदीयमपि वचनाम् प्रतीत्यसमुत्पन्नत्वान् निःस्वभावं ।

madīyam-api vacanam pratītya-samutpannatvān niḥsvabhāvaṃ

[VV, v.22]
(Nāgārjuna's self commentary on the verse 22.)

My speech, because it is dependent on conditions, is contextual (literally 'is without a self-sufficient nature')[7].

Nāgārjuna is saying that the validity of any speech act does always depend on the chosen focus within which I assert. And from this, to prevent oneself from an illusory universal assertion amounts to be aware of the fact that such a formal dialogue can never be finished and, therefore, that there are no proposition in that very sense. And this provides us with a means to understand the famous:

यदि काचन प्रतिज्ञा स्यान्मे तत एष मे भवेद्दोषः ।
नास्ति च मम प्रतिज्ञा ॥

[6] Notice here that the Sanskrit expression for 'thesis' is '*siddha-anta*', 'what is established at the end'.

[7] In this talk, each quotation of Nāgārjuna is from my own translation, taken from my Master Dissertation, *Nāgārjuna et le pluralisme logique*, at the University of Lille in September 2004. I had for this translation mainly worked with the edition of E.H. Johnston and A. Kunst, published in 1978 with the excellent translation of Professor Bhattacharya, see [1].

yadi kācana pratijñā syān me tata eṣa me bhaved doṣaḥ |
nāsti ca mama pratijñā ||
[VV, v.29]

If I had asserted any proposition, this fault (consisting in the paradox of self-reference) would be mine,
but I do not assert any proposition.

We are therefore able to say that when Nāgārjuna says that he does not make any assertion, he is not saying that he says nothing, he is very likely to say that he will not commit himself in the formal process of justification of his positions. The reasons of this refusal are that such a formal proof with a universal claim is vain. A formal proof can in no way be complete. Nāgārjuna's sentence '*I do not assert any proposition*' can therefore be understood this way '*No sentence is 'valid' in the sense of 'universal''*. We can speak together and understand what a given language conveys but we have to keep in mind that this is the conventional level, that at every moment things can be discussed and that the one who wants to reach an indisputable claim whatsoever will be defeated.

At the verse 24, Nāgārjuna is performing a terminological switch from *pratijña* (proposition) to *vāda*. Traditionally, the term *vāda* refers to the philosophical debate or to a claim in a discussion. Now, he have shown that in these lines, Nāgārjuna is performing a speech act that is not fully justifiable. We therefore propose to render this act by the term '*position*' in the sense that it is something that depends on hypothesis, something that is still disputable.

We are going to follow the same line in our understanding of Nāgārjuna's treatment of negation.

2.2 The Art of Making No Negation

A Constructivist Negation... If the negation of a proposition is the assertion of the negated proposition, then this problem also affects the act of negating: the negation of a proposition is always questionable. The problem here is that Nāgārjuna can not firmly establish his criticisms if they are in the negative form.

In verses 61 to 63, Nāgārjuna explains that negating a thing involves the propositional attitude he wants to get rid of because the act of negating is but the act of asserting the negated thesis. Hence, he has to say:

प्रतिषेधयामि नाहाम् ।
प्रतिषेधयासीत्यधिलय एष त्वया क्रियते ॥

pratiṣedhayāmi nāhaṃ |
pratiṣedhayāsi ity adhilaya eṣa tvayā kriyate ||

[VV, v.63]

I do not negate anything,
You foolishly calumniate me when you say 'you negate'.

Now the question remains: what sort of speech act is he performing then?

अत्र ज्ञापयते वागसदिति तन्न प्रतिहन्ति ॥

atra jñāpayate vāg asad iti tan na pratihanti ||

[VV, v.64]

Here, the speech makes it known as false, it does not negate.

In other words, the attack of a given thesis does not lead to the assertion of the negated thesis, but leads to show that the assertion of the thesis is faulty. What Nāgārjuna performs here is an other type of speech act which does not imply a propositional attitude as the assertion does. We will call this act a DENEGATION. Characterizing this act is the goal of the following section.

...and the Operator of Denegation. In the Dialogical approach of logic as introduced above, Keiff developed in [6] a negation which encodes a very similar process.

First of all, what is at stake is to understand a type of negative speech act as the indication of the failure of an act of assertion. As such, this is a constructivist-like negation according to which 'non A' is to be read 'there is no correct proof of A' and not 'there is a correct proof of non A'. What is more, unhappy with the standard way to encode this reading in a formal proof, that is to say unhappy with the interpretation of 'non A' as 'A entails a contradiction', Keiff makes a step that will help us here[8]. More precisely, he develops another reading in which 'non A' behaves like an OPERATOR OF DENEGATION and has to be read 'if you assert A, I will show you that your formal proof of A is not sufficient'.

Now, it is evident that this sticks to what is at stake in Nāgārjuna's approach on theory of assertion when he points out the fact that the signification of an assertion is never unchallenging data. Here, it is worth mentioning that Nāgārjuna makes use of *reductio ad absurdum* arguments, but never uses them in order to establish the opposed thesis. He always uses them in order to show that the attacked thesis is no a justified thesis and that it does not hold. Moreover he does not, in the *Mūla-madhyamaka-kārika*, develop a whole meta theory about the fact that every thesis can be questionable, but he takes one by one every metaphysical thesis in order to show how they can each be disputed.

3 Nāgārjuna and Buddhist Practice

3.1 Dialogical Conclusions and the Everyday Life Strategies

To summarize, Nāgārjuna is saying that we are performing only unfulfilled assertions (respectively negations). Speech acts are never assertions (negations) in

[8] His motivations were different,since he aimed at introducing a notion of relevance that could not be encoded within this standard interpretation.

the strict sense but positions (denegations). The reason for it is that to assert a proposition in a philosophical discussion is to commit oneself to give the justifications for this proposition in such a manner that it will be unquestionable whoever the interlocutor may be while we have to be aware of the fact that the signification and validity of any sentence have their roots in a net of conditions that cannot be entirely enumerated: an assertion (negation) is always still dependent on a hypothesis that is not justified.

Hence, my claim is to say that Nāgārjuna does not call into question the fact that a set of statements expressing epistemically guaranteed beliefs can possibly be considered as a set of statements expressing 'knowledge' (which would have been a skeptical position). What he does question is rather the origin of this guarantee. According to him, the epistemic guarantee is not the agreement between my set of beliefs and reality, but the agreement between my set of beliefs and the successful practice of a community (which is more like an anti realist position in line with Brandom's inferentialism).

In this reading, the Dialogical approach is useful because (in addition to its dialogical frame directly able to express Nāgārjuna's position) it considers a content of knowledge as the practice of an epistemic agent. This is the idea captured by the fact that 'a proposition is the form of an achieved dialogue' or, in other words, by the fact that 'a proposition is the set of its conditions of assertability by a speaker'. Moreover, the Dialogical approach provides with technical tools to express this conception of a proposition, and to express the fact that if the agreement is to be between my set of beliefs and the successful practice of a community, then there can be several distinct types of agreement. And this is what we are going to develop in this section.

First of all, from the Dialogical perspective, Nāgārjuna's claim amounts to the following claim: 'EVERYTHING IS FALSIFIABLE'. Which is not the same claim as '*everything is false*'. It is important to keep in mind the asymmetry between the two players of a linguistic game as captured by a Dialogical game. More precisely, the position of Nāgārjuna could be reformulated in this frame by saying that the Proponent can never have a winning strategy, whether she asserts or denies something. Only the Opponent can have a winning strategy and he always has. There is no achieved dialogue, no more is there a proposition because there is no form of an achieved proof. More formally, let us consider a consequence relation $\models$ (extension to a syntactic derivability relation is straightforward). To say that $\models$ is trivial usually amounts to say that for any well formed formula φ and ψ, $\varphi \models \psi$[9]. But one could also define a dual concept of triviality, namely that for any φ and ψ, $\varphi \not\models \psi$.

If one is to take seriously Nāgārjuna's claim that no assertion is possible, then one cannot escape the conclusion that the logic he advocates is trivial in the second sense, i.e. nothing can successfully be defended against all possible criticisms, not even logical truths for there is not any.

While the whole logic does not seem to allow for a lot of fruitful developments, a fragment of it, namely the fragment in which the Opponent choses to play

[9] Equivalently, if φ is T ('top'), this means that any ψ is valid.

within a sub class of models (a focus), will do. More precisely, we here deal with a sub system in which there are validities. One can even play within classical rules[10]. The only restriction here is that there are no ultimate validity, that is to say, there are no validity against an Opponent that plays the best possible moves. In other words, the Proponent can never have a winning strategy *against an Opponent who plays the best possible moves.* But he can perfectly win against a not acute or comprehensive one. This is why we do learn and communicate in everyday life (*vyavahāra*). The Proponent can win, but he will manage to do so only against an Opponent that grants him concessions. These restrictions are *ad-hoc* (conventional) and they do define a certain type of Opponent. Nāgārjuna does not develop this, but the Jainas will do in their *naya-vāda*, their 'theory on perspectives', in which each set of restrictions of the Opponent will represent a given Indian school of thought.

3.2 The Art of Making Indications

We have been until now explaining Nāgārjuna's position, but we have not yet come to the defense he performed against the ones who argue that the position that everything is disputable is itself disputable. In order to have another approach to this criticism, I would like to come back to the asymmetry between the Proponent and the Opponent in the Dialogical approach. The Proponent is the only one to have COMMITMENTS when asserting a proposition. Everything 'asserted' by the Opponent, is 'asserted' at the meta theoretical level. More precisely, the status of his pseudo-assertions and pseudo-commitments is nothing else that the indication of what the Proponent needs to justify what he is asserting.

Nevertheless, Nāgārjuna is 'protected' from asserting just as long as he is not trying to defend as a thesis the positions he has in his *Vigraha-vyāvartanī* because it is evident that the paradox is present whenever what we are performing at the meta theoretical level is played within the propositional level. Nāgārjuna's 'I do not assert any proposition' is precisely the recognition of the fact that if he had put his meta theoretical positions within a discussion for getting justifications, they would have been challenged. Hence, he is not playing them within a philosophical disputation, but he is taking the only means he has, namely practice. Through this practice, he can show for any thesis in a discussion, that this thesis cannot successfully be defended against all possible criticisms, which is the task of the *Mūla-madhyamaka-kārika.* But this is important to keep in mind that doing so Nāgārjuna is playing at the 'object language level' in a very poor sense: he only takes the pseudo-interlocutor role of the Opponent and, in each situation, shows how to falsify a sentence (which is certainly not the same thing as trying to establish a negative sentence!). I said 'in a very poor sense' because this role is but the indication of a metalanguage position. The same way, in his *Vigraha-vyāvartanī*, Nāgārjuna stays at the level of the metalanguage to

[10] This is a counterargument to the thesis according to which Nāgārjuna holds para consistent thesis.

indicate that he will not go at the level of the language object for any (positive or negative) sentence[11].

3.3 Buddhist Soteriology

The question is therefore the following: how powerful is the act of Nāgārjuna when he is pseudo-asserting? To answer this, it is useful to keep in mind the deep thesis of John Woods, in [12], according to which a 'fallacy' is not a fault of reasoning. This is rather a reasoning such that there is no best reasoning for men, that is for rational agents with limited capacities.

This thesis takes on a new meaning in this Indian theory. Here, the speech act fails to be an assertion. It is but a position in a debate and holding such a position is not considered as a fault of reasoning: it is the right and fruitful way to conduct a reasoning. Speech acts are useful even if they are not propositions. More precisely, language is useful for one to posit himself or somebody else within another web of conditions, within another perspective from which she will be able to notice and signify other things. Language as a mere conventional activity is useful to posit the interlocutor within a perspective in which she will be able to experience useful things for her emancipation. In this argument, Nāgārjuna uses the parable of an artificial man:

निर्मितकायां यथा स्त्रियां स्त्रीयम् इत्य् असद्ग्राहम् ।
निर्मितकः प्रतिहन्यात् कस्यचिद् एवं भवेद् एतत् ॥

nirmitakāyāṃ yathā striyāṃ strīyam ity a-sad-grāham |
nirmitakaḥ pratihanyāt kasyacid evaṃ bhaved etat ||

[VV, v.27]
(Nāgārjuna's self commentary of the verse 27.)

<What I am doing with my speech> is as if an artificial man would prevent from the wrong perception of a man <who would believed> 'this is a woman' where there is an artificial woman

Now, the fault occurring at the semantic level is present at the practical level too : if the assured character of an assertion is something debatable in the object language and if the metalanguage can be put and played within the object language, then it is the target of the same objections and we can not know with certitude this metatheoretical fact, for example, that we are *progressing* in our knowledge.

Our proposal here is to say that this is precisely why we have to practice. More precisely, we can not assert in a *propositional* way that we are progressing, and it is important not to do so, but we can *experience* it.

That is why Nāgārjuna do not go further in its theory of assertion. Following the example of the Buddha, remaining silent on metaphysical questions is a

[11] Here we can think about Tarski's work in which he explains that the object language is strictly included within the metalanguage precisely because of such situations, [10].

crucial step for him to undergo[12]. In conclusion, our claim is that even if it is not developed, the technical and philosophical consequences of such a position are:

- A theory of assertion as act of commitment ; and a theory of the forces of the assertion (assertion versus position, and negation versus denegation).
- To redefine the role attributed to logic. More precisely, it seems that what is at stake is the transition:
  - From a vision of inference (inductive and deductive) as what legitimates the fact that a proposition is considered as an assured knowledge
  - To a vision in which argumentation has the pragmatic function to validate some inferences *in relation to a given perspective.*

References

1. Bhattacharya, B.K.: The Dialectical Method of Nāgārjuna: Vigraha-vyāvartanī. Motilal Banarsidass Publishers, Delhi (1998); (first ed.1978)
2. Brandom, R.B.: Articulating Reasons: An Introduction to Inferentialism. Harvard University Press, Cambridge (2000)
3. Dummett, M.: The Logical Basis of Metaphysics. Harvard's university Press, Cambridge (1991)
4. Garfield, J.L.: Dependent Arising and the Emptiness of Emptiness: Why Did Nagarjuna Start with Causation? Philosophy East and West 44(2), 219–250 (1994)
5. Girard, J.Y.: Linear Logic. Theoretical Computer Science, vol. 50, pp. 1–101 (1987)
6. Keiff, L.: Le Pluralisme Dialogique: Approches Dynamiques de l'Argumentation Formelle. Ph.D-Thesis, Lille (December 2007)
7. Lorenz, K., Lorenzen, P.: Dialogische Logik. WBG, Darmstadt (1978)
8. Rahman, S., Keiff, L.: On how to be a Dialogician. In: Vanderveken, D. (ed.) Logic, Thought and Action, pp. 359–408. Springer, Dordrecht (2005)
9. Siderits, M.: Nāgārjuna as antirealist. Journal of Indian Philosophy 16(4), 311 (1988)
10. Tarski, A.: Logic, Semantic, Metamathematics. In: Corcoran, J. (ed.) Logic, Semantic, Metamathematics, 2nd edn., Hackett (1983)
11. Waldo, I.: Nāgārjuna and analytic philosophy (2). In: Philosophy East and West, vol. 28(3), pp. 287–298. Hawaii's University Press, Hawaii (1978)
12. Woods, J.: Paradox and paraconsistency: Conflict resolution in the abstract sciences. Cambridge University Press, Cambridge (2002)
13. Wittgenstein, L.: Philosophical Investigations. G.E.M. Ascombe (transl.). Blackwell, Oxford (1958)

[12] This metaphysical Buddhist silence of Nāgārjuna is close in spirit to Wittgenstein's silence in L.Wittgenstein, in [13].

Author Index